BACHELOROPPGAVE:

DESIGNING, IMPLEMENTING AND DOCUMENTING THE ATLAS NETWORKING TEST-BED.

FORFATTER(E):

Hans Åge Martinsen

Dato:

19.12.2012

SAMMENDRAG AV BACHELOROPPGAVEN

Tittel:	Designing, Implementing and Documenting the Atlas Network Test-bed.	Nr. : Dato : 19/12/12
Deltaker(e):	Hans Åge Martinsen	
Veileder(e):	CERN, Brian Martin	
Oppdragsgiver:	HiG, Erik Hjelmås	
Kontaktperson:	Hans Åge Martinsen, Brian Martin	
Stikkord	Networking, Test-bed, Laboratory, CERN, Atlas, Environment monitoring	

Antall sider: 130	Antall bilag: 6	Tilgjengelighet (åpen/konfidensiell): Åpen

Kort beskrivelse av bacheloroppgaven:

Atlas prosjektet hos CERN i Genève er et produksjons miljø. For å kunne utvikle nye arkitekturer, teste nytt utstyr og evaluere nye teknologier så trengs et godt støttet test miljø. Dette dokuementet beskriver først forarbeide gjort av meg for å opparbeide forståelse og kunnskap om data sentre og behovene til test miljøet som har blitt opprettet av Atlas hos CERN i løpet av året 2012. Deretter kommer en inngående beskrivelse av prosjektet jeg har hatt ansvaret for i forbindelse med å evaluere maskinevare, innkjøp og utrulling av maskinvaren i laboratoriet. I forbindelse med utrullingen av maskinvaren så ble det utviklet en SNMP motor for å innhente data fra et sett med sensorer for bruk i eksisterende trend-analyse verktøy i Atlas.

THESIS ABSTRACT

Title	Designing, Implementing and Documenting the Atlas Network Test-bed.	Nr. : Date : 19/12/12
Participant(s):	Hans Åge Martinsen	
Supervisor(s):	CERN, Brian Martin	
Employer(s):	HiG, Erik Hjelmås	
Contact person:	Hans Åge Martinsen, Brian Martin	
Keywords:	Networking, Test-bed, Laboratory, CERN, Atlas, Environment monitoring	

Number of pages: 130	Number of appendices: 6	Availability (open/confidential): Open

Short description of the bachelor thesis:

The Atlas project at CERN in Genève is a production environment. To be able to develop new architectures, test new equipment and evaluate new technologies a well supported test-bed is needed. This document describes my initial work to gain an understanding and knowledge about computer centers and the requirements of the test-bed that has been erected at CERN by the Atlas project.

The document then goes on to describe my work done to evaluate hardware, purchasing and deployment of the acquired hardware in the test-bed. While deploying the hardware an SNMP engine was developed that will obtain data from a set of sensors to be used in existing trend-analysis tools used by the Atlas project.

Bachelor's Thesis
Bachelor of Science in Computer Science
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2012

Author:
Hans Åge Martinsen

Employer:
Erik Hjelmås

Supervisor:
Brian Martin

Designing, Implementing and Documenting the Atlas Networking Test-bed.

This document was compiled on December 19, 2012

Contents

List of Figures

Chapter 1

Summary

The A Toroidal LHC ApparatuS (Atlas) experiment at the Large Hadron Colider (LHC) in European Organization for Nuclear Research (CERN), Geneva is a production environment. To develop new architectures, test new equipment and evaluate new technologies a well supported test bench is needed. A new one is now being commissioned and I will take a leading role in its development, commissioning and operation.

This thesis will cover the requirements, the implementation, the documentation and the approach to the different challenges in implementing the testbed.

I will be joining the project in the early stages and start by following the work that my colleagues are doing and then, as I get a better understanding, more responsibility will be given to me.

To be able to suggest and implement solutions I will have to understand what the requirements are and how to achieve these requirements with the given resources.

Chapter 2

Data Centers

Most companies are dependent on computer driven services in one way or the other. To provide these services there will often be a data center, rented from an external company or hosted by the company itself.

A walk through of a generic data center and how it operates will be given, after which Atlas's data center in Atlas Trigger and Data Acquisition Room (SDX) will be explained so that the requirements for the ongoing building of the laboratory is better understood.

2.1 The Generic Data Center

Computer centers contain a large number of servers, networking equipment, power supply and more, all of which need a solid and robust infrastructure to do its work properly. Storage, power, cooling and network are an essential parts of a generic data center and the infrastructure needs to provide for this.

The data center itself is one or several rooms which preferably has sufficient space between the floor and roof to enable the use of a false floor (plenum) if wanted, head-space so that tall equipment like racks can be tilted when moved in and out of the room and abundant room for air to circulate. In addition to space, large amounts of power and cooling are needed to power the machines installed and then to remove the excess heat produced by the equipment.

2.1.1 Power

Abundant amount of power is needed to provide the services inside a data center and if it ever fails, the whole center will also fail which is not desirable. To prevent this, most data canters will have Uninterruptible Power Supply (UPS) (see 2.2.3) installed and also a backup petrol/diesel generator if the cost is deemed acceptable.

Backup Power Generator

A backup power generator is an engine that provides backup power in case of a power shortage. It is able to carry the full load of the computer center and it will wait in the background, checking if there is power, if not it will self-start and provide power for the system until the power comes back or it runs out of fuel [21][19].

When designing a highly available data center, the engineers consider the economical limitations and historical data on power failures in the local area. Is the alternative cost [8] of getting a generator that can supply the whole center with power less than the cost of losing business for the expected down time per time period? Depending on their findings a new generator may or may not be installed.

Note that without UPS (see 2.2.3), the system will still be without power for a couple of seconds until the engine is able to start and begin to provide electricity. Also all the hardware must boot up again, so unless the machines support auto boot up after power failure, the machines will still have to be manually started again. A combination of a UPS and power generator can make the system completely redundant without downtime when there is loss of power.

2.1.2 False floor

Racks are often placed on a raised floor so that there is space under called the plenum which is mainly used as a passageway for chilled air. When using the plenum for this, the Heating, Ventilation and Air Conditioning (HVAC) system must be capable of pressurizing it or the cooling will be inefficient [19]. It can also be used for cable distribution. Doing this makes it less suitable for distributing chilled air as all the cables and potentially changing environment can end up being an obstacle for airflow, making it difficult to

deliver chilled air where it is needed. For this reason some centers prefer to pull their cables above the racks instead. Pulling the cables above the racks also gives the added bonus of making it easier to change out cables in places with a high turnover.

To get access to the plenum and space under the racks, the floor itself is divided into small tiles which can be pushed around or lifted up and removed to give easy access to the space directly under and in the close vicinity of that tile. It is possible to get these tiles in many different shapes and sizes. Mainly the differences are its weight, what material it is made in, the amount of weight they can handle and how much air they let through from the plenum when perforated or grated [19].

2.1.3 Racks

All the hardware that a data center contains are placed in modular racks. Racks can be found in many heights, widths and depths. The industry standard is a height of 42 rack units (Us), but other common sizes are 47Us and 52Us [22]. The external width size can vary, but on the inside one will often find verticals installed so that equipment 19“ wide can easily be mounted. The depth of a rack can vary as well.

For cooling equipment in a rack there is vertical or horizontal cooling. Vertical cooling will get chilled air from the plenum pushed in from the bottom and the heated air will exit from the top. Horizontal cooling gets chilled air pulled in from the front of the rack, through the equipment and out either through the top or through the door on the back.

Vertical cooling is required for open chassis equipment with vertical blades which is the case for several instrumentation standards such as Versa Module Eurocard (VME) or compact Peripheral Component Interconnect (cPCI). Servers however are all front ingress rear egress and require horizontal cooling.

Racks are installed in rows and bolted together for added stability. When racks are being installed, side walls and seals around the border of that wall are installed. This is done to minimize vibrations spreading through the connecting racks, restrict fire propagating, increase efficiency and to minimize air traveling between the racks [19]. Each end of a row is capped off with finished end panels.

When everything is set up and the rack is populated with equipment, the blanking panels go in between installed equipment at the front of the rack to

plug any holes and optimize airflow through the equipment in that rack[1].

2.1.4 Cable Management

Once the racks are bolted down, the inside walls of those racks need cable ladders. One side for power cables and one side for data cables. With these in place the internal rack power distribution is installed just as in SDX.

Signal and power cables should be separated from another if possible to prevent noise and crosstalk. When fastening signal cables, Nylon cable ties should not be used. These can be tightened to high levels and can crunch/deform the cables, increase cross talk and change the impedance of those cables. Velcro straps, which do not have such a high breaking point, but still strong enough to keep cables in check can be used instead [13]. If there are excess cable lengths, these should be stored under or above the rack instead of in the rack itself until it can be exchanged with one that has the correct length.

When pulling cables in the plenum between two places, raceways (see fig. 2.9b), trays and similar should be used to maintain cable integrity [13]. When ever a cable has to be bent to get it to its intended place, those bends should respect the minimum bend radius to maintain signal integrity for copper or mode propagation for optical. Typically this is never less than 5cm [25].

If a cable can not be placed under the floor, measures need to be taken to ensure that data cables are kept away from fluorescent equipment. These are a major source of electrical noise and will make that cable more vulnerable to transmission errors [13].

Labeling

Network cables may be color coded and must be labeled. This could be with a number or bar-code which can be used to reference a data base entry or it could be a human readable label supplying information on what racks it is connected to, which U in each rack the cable can be found in and what port in that U it is connected to [19][13]. Obviously the same label needs to be on both ends of the cable. This is to make it easier to find the cable at a later time.

[1]Statements gathered from training/conversation with Brian Martin when installing new racks in Atlas data center.

2.1.5 Monitoring and Security

All computer centers need to be monitored. Equipment has physical requirements to run as it should and if those requirements are not met then the machines might start performing badly and in worst case scenarios they will break down.

Environment Monitoring

Temperatures rising above or below nominal working temperatures, humidity, condensation temperatures falling below room ambient and doors kept open for too long are all things that needs to be monitored closely so that efficiency can be upheld and downtime or disasters can be avoided.

There are many ways to implement monitoring of the environment in a data center. Typically there are a net of sensors linked to a redundant monitoring service which continuously polls and records the environmental statistics. Rules for acceptable values are set and if exceeded will, depending on severity, issue warnings to operators, send emails issue Short Messaging Service (SMS) or even call the security services.

Security

Physical access to the data center should be controlled and monitored closely to make sure that only people with the proper permissions have access to the equipment. There might be people with bad intentions or who do not have the proper experience to work with the equipment and should therefore not be given the opportunity.

2.1.6 Documentation of inventory and connectivity

The equipment inventory and network connectivity inside the data center needs to be documented somewhere. A solution to give the users of the center an easy and scalable way of storing equipment and network information so that it can be looked up at a later time when needed is a requirement. There are many tools for this in varieties of complexity depending on size and numbers of the data center.

2.2 Atlas's Data Center

The laboratory being constructed in building 4 is being built from scratch and many of its requirements, features and practices originate from the implementation of the Atlas Trigger and Data Acquisition Room (SDX) which is in a building at the top of the shaft above the underground experiment. Data is fed to it as Ethernet traffic. This data is processed in real time and the filtered results output as Ethernet traffic. It is therefore just a data center. The requirements for it were laid out in the Trigger, Data and Acquisition Technical Design Report[4].

While Atlas was under design it was expected that it would need 2500 or more dual core Central Processing Units (CPUs) running at 8GHertz (Hz) each, which in 2003 was the expected availability of CPUs in 2007 when the project was scheduled to start taking data.

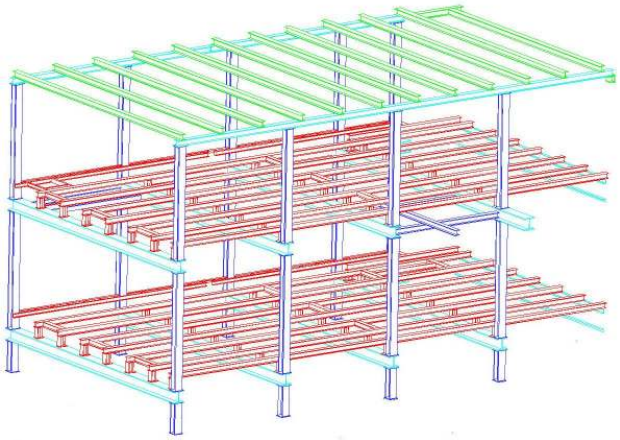
Relying on Moores law to provide the required power over the intervening time, and knowing that it would take time to reach the design energy of the LHC, it was decided to construct a building to house about 2000 servers. From this basic number it was possible to extrapolate "typical" numbers like physical volume, power consumption and the weight that this would represent.

For Trigger Data Acquisition (TDAQ) purposes the design was centered around network booted servers with a single hard drive. In general these machines have been mostly 1U, sometimes 2U form factor, weighing in at around 16kg, up to and sometimes above 20kg if redundant power supplies are used. Power usage has been increasing, but stabilized at around 300Watt (W)/U for the type of machines used by Atlas.

To support 2500+ CPUs, it was decided that SDX would need 3000Us of rack height, each U consuming ~300W and weighing in at ~20kg.

Racks designed for industrial use are typically between 42 (ref. 2.1.3) and 52Us. For a weight and power approximation we can assume that a rack is 50U. In this case, a rack that is fully occupied will represent $50 * 20kg = 1000kg$ which is hard to sustain in terms of floor loading. Equally, the same rack fully occupied will represent $50 * 300kW = 15kW$. Cooling this is possible using air cooled systems like warm-aisle cold-aisle techniques (see figure 4 for illustration [18]), but requires large air volumes around the aisles to achieve the desired effects. Water as a cooling medium can help the situation, but is also restricted to around 10kW/rack if the racks are on a 60cm pitch. To meet these limitations, Atlas chose a practical limit of ~30

Figure 2.1: *Steel girder structure hosting Atlas's surface data center[5].*



servers/rack. Fully equipped this would represent <600kg and <10kW per rack. This increased the number of racks needed to 100 when taking into account the required overhead.

2.2.1 Volume and Weight

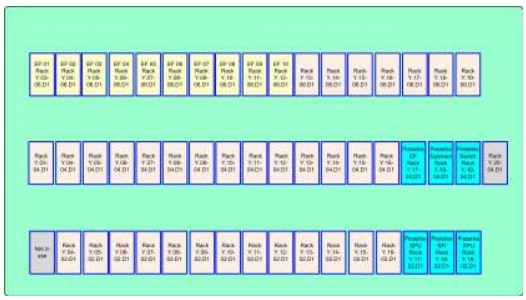
The design of the data center is reflected by the requirements made by the fact that it need to host 100 racks and that the space in all directions in its current location is limited by the fact that it is placed behind the elevator shaft going down to the LHC tunnel in SDX.

The figure 2.1 shows the steel structure installed to support the data center. Each floor has three pairs of girders, each pair of girders have racks placed directly on them. This way, a load of up to 750kg per rack is possible, giving Atlas a little headroom in terms of weight and allow up to 17 racks with walking space at both ends of each aisle. To accommodate door openings and similar some allowances was made and the final placement is shown in figure 2.2.

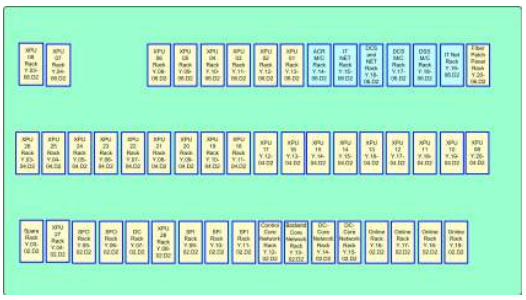
Although it was decided to only use 30 of 50Us on average, the infrastructure must still be able to provide for racks being fully equipped. This is because future technology might provide solutions that make the data center able to meet the weight and power restrictions with fully occupied racks.

Figure 2.2: *Lower and upper level of SDXs data center[12].*

(a) Lower level SDX



(b) Upper level SDX



2.2.2 Power Distribution

The power distribution in SDX was designed so that a three phase protected feeder circuit provides power to individual branches, each branch supplying 230Volt (V) rated at 16Ampere (A) which is also the maximum power rating for end equipment cables and power plugs. With an expected power usage of ~300W, 10 machines can be fed without over subscribing a branch and leaves some space for headroom (368W/machine). Using all three phases, 30 connectors can be provided which is still not enough to feed the expected servers or potential number of "lower power" devices to be installed in each rack.

When using 3 phase power, the load must be balanced over the three phases so that one phase is not overloaded. For this reason the logical step up is not to add a branch, but to double the number of branches using two feeds to permit a better load sharing. With 6 branches in each rack one could now populate each branch with 8 sockets instead of ten, giving a total of 48 sockets per rack and further increasing the headroom from 368W to 460W per U (machine).

This means that a rack can now consume up to $230V * 16A * 6_{phases} = 22kW$ without tripping the circuit breakers in the rack, far more than the initial predicted <10kW per rack. This "overuse" can be accepted provided that racks in the close vicinity compensate for this by having less heat output than nominal. To make sure that over subscription is not done, groups of 5 racks are supplied from a 100A three phased circuit breaker, which means that any group of 5 adjacent racks can use a maximum of $230V * 100A * 3_{phases} = 69kW$ without tripping the circuit breaker. This value is close enough to the target figure of 10kW/rack.

Each row, seen in figure 2.2, must have four of these groups to be able to provide for the nominal 17 racks per row. So, for each rack there are two feeds of 16A 3 phase. These are shown in figure 2.4a as 1,2 up to 9,10. The manual breakers that feed these 3 phase feeds are connected to a remote controlled breaker which can turn the power on or off by the use of a Programmable Logic Controller (PLC). This is controlled through the Detector Control System (DCS) system and is shown in figure 2.4b. Each row feeder and its four group feeders are housed in a wall mounted cabinet and for each floor there are three of these cabinets. This solution meet all power distribution requirements except for one challenge.

The power supplies of rack and server type hardware are often subcontracted out by the vendor to one or more third party suppliers. The vendor will provide specifications for most things, but seldom for the turn on surge current. This initial surge can, for a very short period, be many times the maximum nominal current and can last for several milliseconds. Tests done in SDX showed that the in-rush current could be as high as 80 times nominal current. When turning on a fully loaded rack, the aggregate surge will then trip most circuit breakers and differential breakers. To get around this, D curve circuit breakers were installed in series with thermistors in all racks. The thermistor has a high resistance while at nominal room temperature and as it heats up from the current passing it, the resistance will go down[23]. Cold-start in-rush current is thus limited to <90A per phase for the few milliseconds of overload. The D curve trip response time for a factor of 6 ($90A/16A = \sim 6$) is >1second and thus the trip is avoided.

Using thermistors solved the over current challenge, but not the differential current problem so there are no differential breakers installed in the racks. To avoid personal hazard it was decided that consumer equipment compatible plugs and sockets are not to be used, but instead employ "Eco-Bus" connectors (see figure 2.4a).

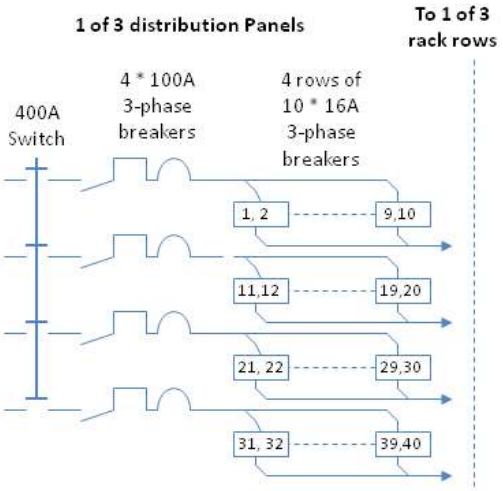
The two 3-phase 16A feeds per rack are brought to a repartition box mounted in the lower 3Us of the rack. The box contains manual circuit breakers so that the rack can be isolated for maintenance purposes. It also houses the in-line thermistors plus the PLC breaker for remote control.

The circuit diagram for one row is shown in figure 2.3a. It covers the distribution which can be summarized as follows.

- One rack maximum: 22kW
- Five racks maximum: 69kW

Figure 2.3: *Two figures describing SDX's repartition boxes and distribution panels[12].*

(a) 400A Distribution



(b) Per Rack Distribution

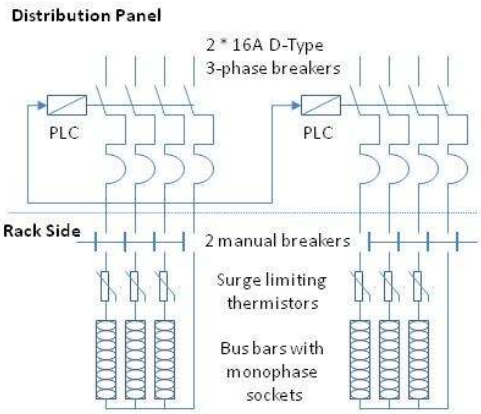
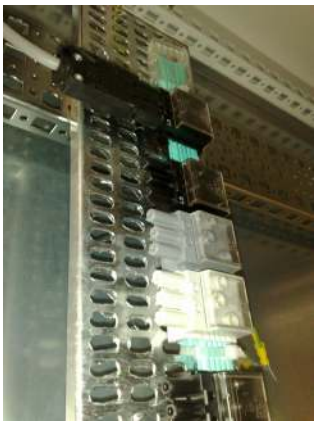


Figure 2.4: *Eco-Bus and overview of the cables coming in and out of the repartition box in the lower three Us of the rack.*

(a) Eco-Bus, each color tells you which phase it is connected to



(b) Breaker box with cables

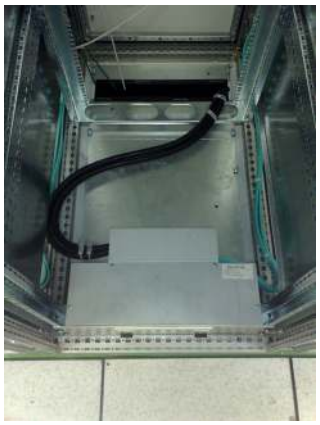


Figure 2.5: *Atlas's UPS*

(a) Front

(b) Inside hall

- One row maximum: 276kW
- One floor maximum: 828kW
- Building maximum: 1.656MW

There are also critical services installed in SDX like for example DCS which must always be powered on even if the main power distribution fails for any reason. For these services an UPS had to be provided (see figure 2.5), which also made it possible to increase availability in less crucial but important services like the core network, attached storage and server infrastructure. Having these on UPS reduce the delay in resuming services after an interruption.

2.2.3 Uninterruptable Power Supply

SDX has both diesel backup and UPS installed locally. The UPS installed in SDX (see figure 2.5), is able to provide power for the critical services until machines can be shut down safely or until the power generator is able to start and take over the load.

The UPS provides its service at all times since in addition to providing availability for services, it also filters the incoming power for spikes and drops, protecting the on-line equipment from these effects[24].

2.2.4 Cooling

SDX can be considered as a closed box. If power is fed into the box, we must assume that the heat converted from this power must also be removed. Without proper cooling most data centers would be rendered useless in a short time.

Main Water Cooling

Space is a major constraint in SDX. Unlike general data centers that are often "green field project" constructions, these space constraints force Atlas to use water cooling techniques to exchange the heat generated [26]. 90% of all heat is cooled with water, the remaining 10% is cooled with air. Each rack has a water-cooled rear door-mounted heat exchanger. The water passes through pipes that are bonded to cooling fins through which air is drawn (see figure 2.6a), by a series of 3 (or 5 depending on the model) radial turbines (see figure 2.6b). The heated water is then transferred out of the system and reused after being cooled down to nominal temperature again [5]. When the heat exchangers are fed with ~$2m^3/h$ water at optimal temperature together with optimal air circulation and temperature, the 3 turbine heat exchanger can cool 9.5kW and the 5 turbine version will cool 14.4kW. See page 103 for technical details on the heat exchangers.

Both inlet and outlet water hoses are connected through spigots to the water distribution system. Each spigot is fitted with taps so that it can be opened or closed when doing maintenance.

Each branch distributing water to each row has a flow of $20m^3/h$, which means that each of the 17 racks get just above $1m^3/h$ of water which is clearly less than optimal. In practice this has been shown to not be a problem. First of all, vendors over specify their power supplies nominal value to prevent problems when the machine is using the maximum amount of power. Secondly, on the lower level, only 50% of the racks are populated.

The currently available cooling in SDX (after running for some years) is at its limits. The branches feeding each row with water have been manually tweaked to provide cooling where it is needed the most at the expense of others. The remaining margin is needed to cope with the worst case ambient during the hottest summer days. The plan is to upgrade the water supply system feeding SDX to allow fully populating of all racks in SDX.

Figure 2.6: *Rack doors with heat exchangers and turbines*

(a) Heat exchanger with inlet and outlet hoses

(b) Rack door with 5 turbines

Secondary Water Cooling

Since SDX has UPS installed, the situation that equipment might be running without cooling can happen and has happened (see 2.2.6). If the center for any reason loses the main cooling circuit. Then under normal circumstances, machines will be safely shut down, but there are machines running on UPS which can not be turned off under any circumstances. Examples are the DCS (see 2.2.5) and core network racks in second floor of SDX on row 6 column 15-20 and row 2 column 12-15 (see 2.2b).

These machines will now cause the room ambient to rise since the water is no longer circulating through the heat exchangers and removing the generated heat. To cope with this situation and provide cooling for the racks on UPS in these situations a secondary circuit was installed which uses normal tap water drawn from the lake and dumped to drain after use. The ingress water temperature will typically be 12-16 degrees depending on the season and will be insufficient to contain the ambient level but it will be enough to stop it rising above tolerable levels.

DCS will notice that there is something wrong with the main water circuit and swap to secondary automatically. The valves controlling the whole process can be found on second floor under the tiles between rack 19 on row 2 and the vertical access ladder on the wall.

Air Cooling

The last 10% of the cooling is done by air being circulated through the equipment and building. Chillers in the pump room (building 3182) next to SDX supply the cooling for the forced air circuit.

In addition to providing chilled air, the system will also remove humidity being added by people breathing the air and from outside air being added to the existing air. The process of removing humidity is expensive in energy, since the air must be chilled first to remove water vapour and then heated up for use. It is however necessary to keep the dew-point lower than the temperature of the water coolers to make sure that water does not condense on the machines in the data center.

2.2.5 Monitoring and Security

All the equipment in SDX need to be monitored (ref. 2.1.5). The DCS system provides this service for Atlas. If something goes wrong or is about to go wrong, the people responsible need to be aware of this. To provide this each rack has two measurement probes. One is installed on the back of the heat exchanger inside the rear-mounted door and it measures the air exiting from the middle of the rack. Since there are 3 or 5 evenly distributed turbines pulling air out of the rack and there is an even number of machines distributed in the racks the probe is essentially measuring the same value you would find in the rest of the rack. The heat measured will then be a function of the average load a machine has, the efficiency of air extraction and inlet water temperature at that time. Typically a rack will have an internal temperature of $(40 - 45)$°Celsius (C) depending on the machines being idle or busy. The second probe is attached to the exit water pipe of the heat exchanger and is covered with plastic foam thermal insulation. The inlet water temperature is fixed and is measured by DCS. With efficient cooling the Δt in °C between inlet and outlet water temperature is between 4 to 6 degrees depending on the thermal load of the rack.

Each turbine in the door has a fan fail signal. These are multiplexed and then fed into monitoring so that if any one turbine fails, a message will be given to the people responsible. One turbine failing alone is not a cause for worry, but rather a call for maintenance and replacing of the turbine.

In addition some other aspects are monitored. The ambient room temperature is measured and monitored on both floors using two sensors, one on the

wall for a "typical" reading and one in the ceiling where there is a recognized hot-spot. The ambient room humidity is also carefully measured and monitored in case the dew point rises above the inlet water temperature. If this were to happen, the run-off can be significant and drip down on to electronics on the first floor or 100m down to the pit since SDX is mounted directly above an access shaft. There is smoke detection in the whole center. If smoke is detected DCS will automatically alarm the fire brigade, message the Shift Leader In Matter Of Safety (SLIMOS) and send a SMS to the person on-call at the time.

A last safety service has been added where all eXchangeable Processing Unit (XPU) racks will be shut down automatically if the temperature of the room exceeds 45°C. If for some reason cooling stops working all together, even the secondary one, then experience has shown that the temperature will rise in the data center with about 1°C every minute and will reach critical level in a very short time. DCS will shut down the XPU racks, not to stop the temperature rising completely, but decrease the speed at which the temperature is increasing inside SDX.

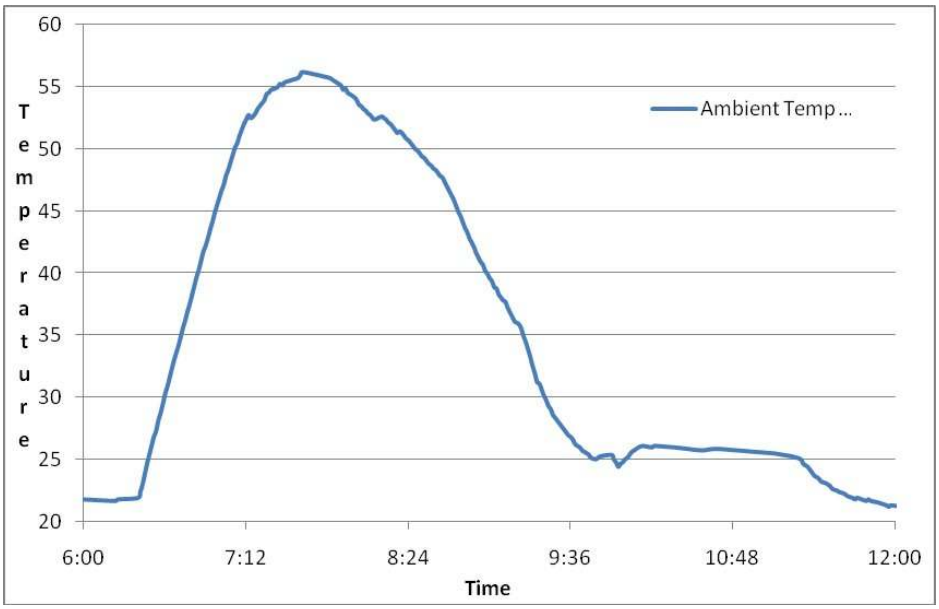
2.2.6 Cooling failure in SDX

On the 26th. Of November 2008 there was a cooling and ventilation failure at SDX which was not brought under control before the temperature rose to excessive levels and services started breaking down.

Early that morning there was a failure in the cooling plant residing at Point 1 which halted the production of chilled water used for both water (rack cooling) and air cooling. This failure also stopped the circulating pumps which meant that all cooling was lost.

The failure was not notified by the cooling and ventilation system and the first indications of anything wrong came as warning messages in the DCS system saying that temperatures were rising in SDX. These warning messages were sent out as SMSs to various responsible people. All of the critical warning messages were not noticed for various reasons and by the time people are noticing that something is going really wrong, the temperature in SDX has risen to 40°C and it has been 25 minutes since the failure started. At some point one of the Sys-admins wakes up and sees the missed SMS messages and tries to remotely shut down XPU racks, by then all except one link from General Purpose Network (GPN) has failed and the last one fails before the person can power down anything. The data center in SDX is now almost

Figure 2.7: *A graph showing temperatures over time while the cooling failure was lasting[7].*



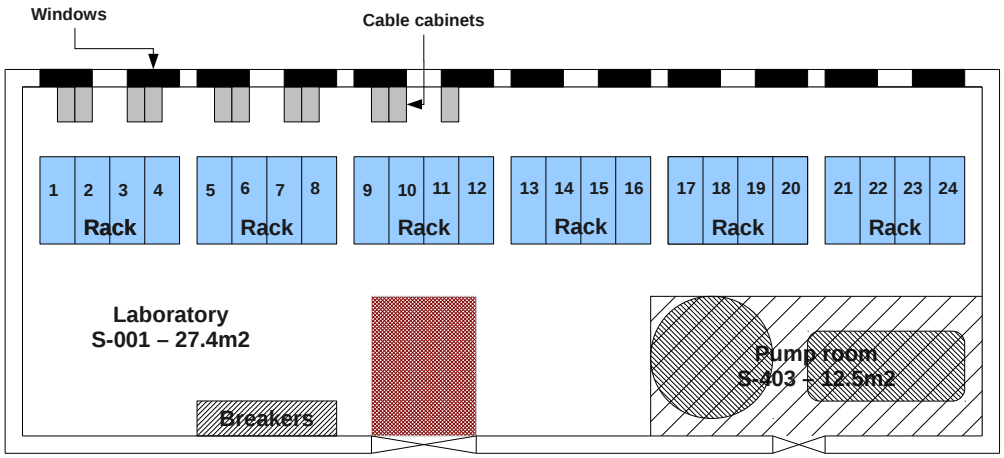
at 60°C and the severity of the situation is now being realized. The fire brigade are called for help plus the SLIMOS are made aware of the severity of the situation. 1 Hour and 13 minutes has passed since the failure started. Shortly after this, all XPU racks are shut down together with the switches. From start of failure to the time of shutdown of equipment it took 1 hour, 21 minutes, the ambient room temperature was measured to be 56.5°C, highest internal rack temperature was 100°C and the PC casings were measured to be between 65°C and 85°C at shutdown.

This incident could happen because trust was put in that DCS would send SMSs to people with warnings and that the receiver would read them. However the monitoring did work but the shut-down had not been properly tested and it did not function. In addition the thought was that if temperatures would rise above critical level, the equipment itself would shut down providing a sort of auto-regulation. This however did not happen because of default Basic Input Output System (BIOS) settings in the equipment installed in SDX. Today, there are systems in place to make sure that temperatures will never rise over 45°C and if they do, racks will be automatically shut down. The people working with or responsible for the system is now taught to check that their telephones are in working order, their SMS buffers have space and that some critical settings are as they should be.

Fortunately the incident happened without big losses and by the end of the day, many of the racks were up and running again, ready to process data for Atlas. The whole incident is documented thoroughly in [7].

2.3 The Laboratory

Figure 2.8: *Overview of the laboratory in building 4*



As mentioned earlier in 2.2, many of the specifications and requirements for the laboratory origin from SDX. The reason for making the laboratory as similar to SDX as possible is to create a physical environment that is similar to that being used in Atlas's production systems.

2.3.1 False Floor

The false floor in the laboratory has been designed to withstand the same weight as the false floor in SDX. Unlike SDX the height of the plenum is

Figure 2.9: *The False Floor in The Laboratory*

(a) Overview of the false floor

(b) One of many cable-trays under the false floor

not very high. In fact the false floor is just high enough so that the water plumbing with insulation around it and spigots are clear of the tiles above it. In figure 2.9a you see the installed false floor in the laboratory.

2.3.2 Racks

The laboratory has been fitted with 22 Rittal TS8[2] racks, and prepared for two more to be installed. 18 of the 22 racks are installed with 5-turbine heat-exchanging doors and 6 with 3-turbine heat-exchanging doors which have been recycled from SDX after an upgrade. Each rack has been fitted and prepared to receive individual 1 U machines with the same specifications and requirements as in SDX in terms of power, weight and cooling.

2.3.3 Power

Each rack has been equipped so that it can meet the exact same requirements as in SDX. The only difference from the original implementation is that the distribution panel has 5 groups of racks instead of 4 fed from one 400A circuit breaker. Since $400A/24_{racks} = \sim 16.7A$ and $230V * 16.7A * 3_{phases} = 11.5kW$ is above 10kW, all racks can still work at maximum estimated load without failing.

[2] Rittal TS8 Data sheet - Last accessed 15th. July 2012

Figure 2.10: *Distribution panel installed in the laboratory*

Scripted Power Control

The same PLC controlled breakers employed in SDX are also used in the laboratory and are housed the same way in the main distribution panel.

In SDX the control of these PLC-driven breakers is done by DCS and it has no support other than the DCS experts at CERN. Also any development for that system would have to be done on Windows which would make it the only Windows requirement in the whole laboratory. For coherence it has been decided to find a Linux based solution.

2.3.4 Cabling

Fully structured cabling has its own stringent rules for avoiding power lines and electrical interference. Some of the these practices can be found here 2.1.4. We don't have the space to conform fully to these rules, but at least we avoid power cables where possible and definitely avoid passing close to any fluorescent light fitting which are known to be a big source of radiation noise.

The cables inside the racks are fastened in the same way as described in 2.1.4. The ladders used in the laboratory for putting signal and power cables can be seen in figure 2.11.

One of the best practices mentioned earlier in 2.1.4 was not followed in SDX which created a situation which was unfortunate. Me and my colleague had to go and look for a cable without labels. In a place with so many labels it can

Figure 2.11: *Ladders for cables*

(a) A ladder for power **(b)** A ladder for data cables

be pretty easy to find both ends of the cable, however it is not easy to know which cable you are to choose if there are more than one cable without labels and the cable goes under the floor. In such cases where a cable without labels have been found and there is no knowledge of where the other end goes, a cable detector can be used to search for it. One end is connected to a device that outputs a known signal. Another device is then used to search the area around bunches or individual cables. You will be given a clear message the closer you come to the cable the signal output device is connected to.

The point is, if order is kept in the cables, it will be easier to identify them. Accidental ripping, trampling and removal of wrong cables are less likely to happen. Also it gives the cables an owner to contact in case of problems or questions.

2.3.5 Cooling

The laboratory has a potential to dissipate heat for up to 100kW if the adjacent laboratory belonging to A Large Ion Collider Experiment (Alice) is accounted for too[3]. Just as in SDX, the choice was made to use water as a cooling medium using rear door mounted coolers.

There is one vital difference between the laboratory and SDX. In SDX the cooling is the experts domain, but in the laboratory it will be the users of

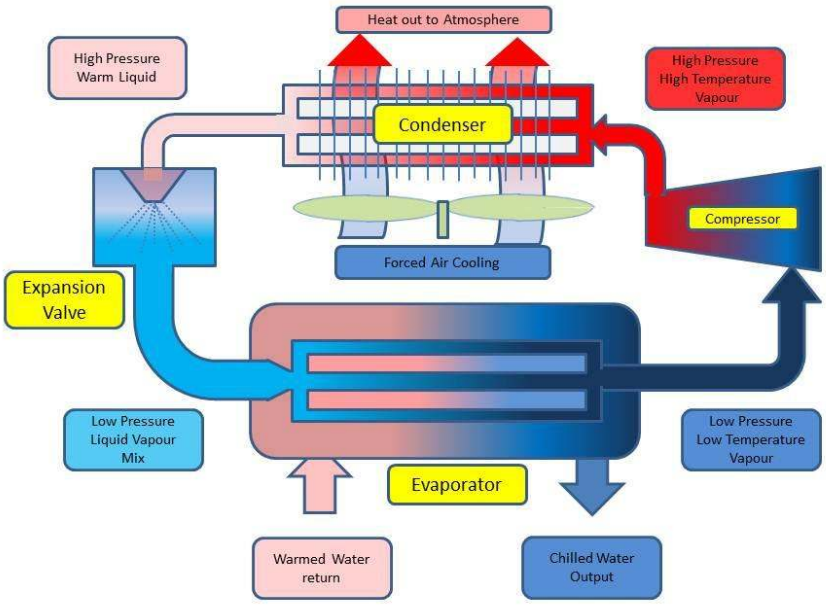
[3]Water cooling in the lab is shared with the adjacent Alice project laboratory.

the laboratory that will have to know what to do in case of problems and therefore they will have to be educated.

2.3.6 Water Circulation

The water cooling system is divided into two separate distribution circuits. One circuit cycles a refrigerant through a thermal cycle using that refrigerants properties to absorb heat from another media when cold and under low pressure. The other circuit containing water, absorbs heat and passes it to the first circuit through a thermal process as shown in figure 2.12.

Figure 2.12: *Refrigerant thermal cycle[14]*



Refrigeration

The first circuit will take a refrigerant, compress it, making it hot in the process. That gas is passed on to a condenser which will cool down the compressed gas and force it to condense into a liquid. The following procedure will abruptly reduce the pressure, making parts of the liquid instantly vaporise. This will cool down the refrigerant down to the boiling point of that

Figure 2.13: *The 30RB Aquasnap chiller which is installed on-top of building 4*

liquid at the reduced pressure.

This liquid is passed through a evaporator which lets the refrigerant be in thermal, but not physical contact with the warm water that has been passed around in the second circuit. The refrigerant will absorb heat from the water, start boiling and pass the heat to a compressor which will start the whole process over again. This whole process is packed into one single machine called a chiller and the one used by the laboratories is a Aquasnap[6]. It is pictured in figure 2.13 and is installed on top of building 4.

The chiller is scaled for 100kW shared between Atlas and Alice. Since the two laboratories together will not be running at this capacity all the time, a water storage tank [F] has been installed to prevent the chiller from being cycled too often and reducing its lifetime. The way it works is that the water flow coming from the chiller [A] on the roof has a slightly higher flow rate ($48.6m^3/h$) than what the pumps inside room S-403 [B] is providing to the laboratories ($45.8m^3/h$)[14]. So while the chiller is running the excess water lowers the temperature of the tank contents. When the temperature of the tank content has been lowered enough the chiller then switches off and the pumps cycle water from the tank instead. At some point the tank becomes too warm to do its cooling purpose and the chiller is switched on again and the cycle repeats.

After the three first islands of racks were powered up and had been running for a while, all gages concerning water temperature were checked.

- Temperature in: 16°C
- Temperature out: 18.5°C

Figure 2.14: *An overview of the cold water distribution in the laboratory[14]*

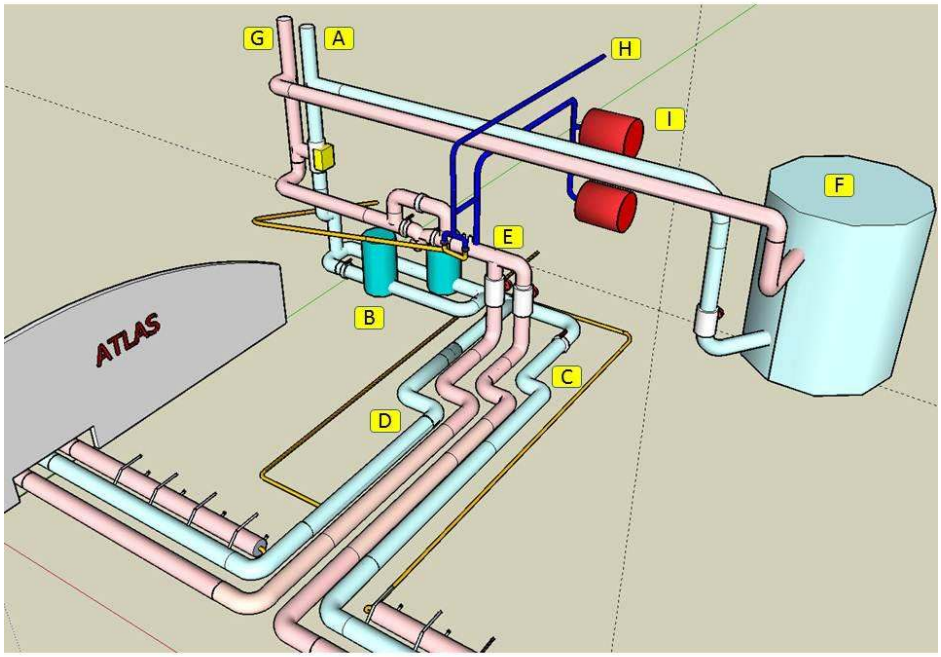


Figure 2.15: *Two pumps and a pressure tank installed and ready to start working.*

(a) Pumps used to pressurize the circuit

(b) Cold water tank which is used for buffering cold water to avoid stopping and starting the roof cooler too often

Figure 2.16: *Daikin HVAC ready for use.*

2.3.7 De-humidification

It has been decided not to install equipment to remove humidity from the air circulating the laboratory. The cost of implementing such equipment would be far too expensive and elaborate compared to its benefits. However, an investment in a system that will monitor the dew point of the ambient air has been made and will be explained more closely in chapter 3 and 4.

A Daikin HVAC was already installed in the room (see figure 2.16) from prior use and is ready to be put in production. Although it does not meet the requirements for the lab, the cost of installing a new and better system that can provide the full requirements is not worth the extra money at this point in time.

Condensation is found in the laboratory

The day after the first 8 racks were powered up, condensation was found in the corner, on top of some machines waiting to be installed. One pipe above these machines were sweating a little. The pipe carries water from another service that is colder by a few degrees than the coolers in the room which is why it is a sensitive point. It is also a good demonstration that under certain conditions the dew point can be very close to the operational temperature of the chillers and will need close monitoring. Because this pipe is actually a little colder than the chillers it will be insulated to prevent dripping. It is being followed closely until it can be covered up with insulating material by one of CERNs staff.

2.3.8 Network

Although it is expected that the use of the laboratory will evolve over time, there are already clear indications of the main activities for the next few years. These are divided fairly evenly among the six 4 rack clusters.

- 01-04: Atlas DAQ testbed.
- 05-08: Racks of servers, can be testbed hosts or traffic generators.
- 09-12: Net-admin testbed.
- 13-14: Sys-admin services.
- 15-15: Read out and trigger testbed.

After a meeting with the IT department, it was decided that the laboratory would be given the private address space[4] 10.193.0.0/16 and that every Internet Protocol (IP) in that subnet would be accessible from GPN but not from outside CERN. The laboratories connectivity to the outside world is a single Gigabit Ethernet (GE) copper connection with a /30 subnet defined for it.

The GE up-link to GPN will be tested to see if it is enough for the laboratory to function. If not, it has been suggested to upgrade the up-link to a single 10Gb fiber instead. This however can create a problem for the rest of building 4 since the laboratory has the potential to saturate such a link and the buildings star point only has one 10Gb up-link to the backbone.

It is also desirable to use the IT departments Dynamic Host Configuration Protocol (DHCP) servers to supply IP addresses to devices in the laboratory as much as possible just as in SDX. This means that any device that needs to be supplied an address from the external DHCP server must be registered in Local Area Network Data Base (LANDB).

Dynamic address control in the network

The laboratory will be using the IT departments and some Sys-admin-maintained DHCP servers as much as possible to assign IP addresses. ITs servers will be providing IP addresses to devices which the Net-admins are responsible for and the remaining servers will provide address resolution for the servers in the laboratory.

[4] A subnet of the 10/8 (A) address space defined in RFC1918

The DHCP protocol has been designed to use the broadcast address on a given subnet to ask any local DHCP server that is listening to provide an IP address. Since the laboratory uses the IT departments DHCP servers to assign IP addresses and these servers are not on the same subnet as the devices itself, the core router in the laboratory must be configured to route any DHCP request to these servers. This is done by using the *ip helper-address* functionality found in the HP core router. With this functionality, any DHCP request will automatically be forwarded to all helper-addresses defined in the configuration for that subnet or Virtual Local Area Network (VLAN).

Security

Since all IP addresses in the laboratory are accessible from GPN, it is very important that all switches and routers in the laboratory have a user name and password set for their management interface. A suggestion would be to put all the management interfaces on a dedicated VLAN which is accessible from a set of subnets (called a service in LANDB) configured to be allowed routing to and from that VLAN on the core router.

2.3.9 Inventory documentation

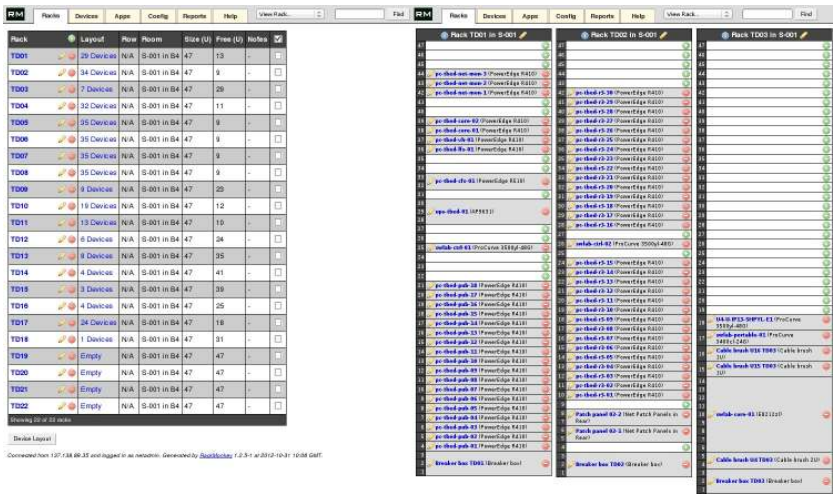
In SDX all inventory documentation per rack is done through the RackWizard[5]. It is not possible to use this software in the laboratory so another solution had to be found. Initially the idea was to populate an excel sheet and put it somewhere all could modify it. This was a stop-gap solution that had limited functionality and is easy to corrupt without traceability by an untrained user. After having searched for open source alternatives, a candidate was found (RackMonkey). The software was installed on a test machine, demonstrated for the team and was approved for deployment.

Some pros of using RackMonkey:

- Relatively simple to set up and maintain.
- The tool is web-based making it easy to access from all Operating System (OS) platforms.
- Gives the end user an interface which is intuitive.

[5] CERNs RackWizard can be found here.

Figure 2.17: *A picture divided in two parts; Left side: RackMonkeys front page. Right side: A visual view of the three first racks in the laboratory.*



- Uses Structured Query Language (SQL) to store data, making it easy to include this service in the laboratories MySQL backup plan.
- Licensed under the GNU General Public License (GPL) version 2 making it free to use and modify as long as the license rules are followed[6].

Some cons of using RackMonkey:

- The tool has not been updated since September 2009 and is no longer in active development.
- The tool is written in Perl which can be a problem if any development has to be made to make it fit the needs of the laboratory since most of the development done in the Net-admin group is done in Python.
- No built in user authentication. RackMonkey suggest using Apache HTTPs simple authentication service for user authentication. It can however when using Apache HTTP differentiate between a special user called "guest" and all the others. This user can not write to the MySQL data base.

After getting input from the users of the laboratory, some changes were made[7] to the RackMonkey source code to implement needed fields in the device section. The software has been released as a service and is in use.

[6]The complete license can be found here.

[7]The changes can be found in the lab4 branch of this git repository.

Chapter 3

Environmental Monitoring Hardware

Many of the laboratory's 22 racks have already been populated with the equipment assigned to it and needs monitoring. In SDX, the environmental monitoring part of the DCS system is used to gain the needed supervision of its equipment ref. 2.2.5. This system will not be available in the laboratory which means another system needs to be found, evaluated, installed and commissioned.

3.1 Requirements

At the time of my induction to the Net-admin team, some work had already been done to identify constraints and needed functionality from the environment monitoring equipment to be used in the laboratory.

The DCS system is complex and made for large industrial environments and is the domain of its respective experts in Atlas. A system like this (sometimes called a Supervisory Control And Data Acquisition (SCADA) system) is too complex for the laboratory needs and requires one or more people to specialize in it. Since it will be the users of the laboratory that maintain and understand the chosen system, a easy to understand, low learning curve stand-alone environment monitoring system will be sought after.

Requirements:

- The ambient room temperature and relative humidity of the data center

should be known. By knowing these values, the dew point of the air inside the center can be calculated[11]. This value must always be lower than the temperature of the door coolers otherwise condensation will form and the computers will need to be shut down. In addition this will give historical data which can be used to evaluate the bounds of normal operation.

- The in-rack temperature must be known throughout the racks in production so that warnings and possibly automatic measures can be executed when temperature hits known important levels.
  - First warning "level": Temperature above nominal temperature.
  - Second warning "level": Close to critical temperature.
  - Third warning "level": Temperature is $> 40°C$ (first instance of devices/equipment like fiber connections stop working [7]).
- The chilled ingress water supply to the laboratory and egress water temperature of each rack must be monitored closely. Any local increase or decrease in ΔT outside of its normal fluctuation pattern in any one rack can indicate increased load on machines in-rack, failing fans, blocked air circulation, accidental or deliberate change of water flow to that rack. A ΔT of 2-3 degrees outside of its nominal working range can be a lot so the equipment monitoring it must be sensitive enough to pick up changes within this range.
- The plenum needs water leakage sensors so that any leakage from the ingress and egress water lines are discovered as early as possible and fixed.
- The system must provide real-time monitoring plus have the option to export that data to external hosts.
- The whole system should be small, easy to understand, easy to maintain and specialized for the purpose of monitoring the environment in a computer center.
- The system should be able to read digital signals (dry contacts) and preferably also send signals.
- The system should be Linux based as most of the Net-admin-teams administration knowledge is based on Linux. If the chosen system is running an OS other than Linux, it will be the only system in the laboratory, administered by the Net-admin team which is not Linux

except from the switches and core router. This would be an additional unwanted load.

- The system should have an option to add additional third party sensors, either through built-in connectors or external devices that can be connected to the system. If this option is present, the already installed rack door and turbine sensors can be used without having to invest in additional systems.

- It is a positive thing if the system is made with redundancy in mind in terms of either built in UPS and/or dual power supply.

3.2 Choices of hardware

The initial research revealed many retailers of these light-weight low-learning curve environment monitoring systems. Some of them specializing on this field only. After studying the alternatives, three product lines from three different distributors were chosen for a closer look.

- NTI http://www.networktechinc.com
- Jacarta http://www.jacarta.co.uk
- AKCP http://www.akcp.com

In common for all the systems is that they are designed for the purpose, rack mountable, support commonly used protocols and provide a user interface for viewing the retrieved sensor data. The big differences between the systems is the amount of accessories, the degree of precision in their sensors, price and user interaction.

3.2.1 NTI

Network Technologies Incorporated (NTI), a company based in USA sells a wide range of products, one of them is the ENVIROMUX-SEMS-16U (EMS16U) environment monitoring system.

Figure 3.1: *Picture of EMS16U. Property of NTI*

The EMS16U has features and accessories that will make it able to provide the laboratory with all of its requirements. Temperature, humidity, water leakage can be monitored with a series of sensor accessories that can be bought from NTI and connected to its Registered Jack 45 (RJ45) ports. Dry contact devices can be connected to the units built-in digital ports making it able to monitor those as well. One unit alone can not provide resources for all the required sensors and therefore has an option to connect several EMS16U in a network by either daisy-chaining 4 units or connecting 6 units through an Ethernet network. To ease management of the network one of the 4 or 6 networked devices can be used to provide a centralized server/user interface. It is possible to connect more EMS16U in one monitored network, but not without buying additional software from NTI to do so.

In addition to the described ports the unit has many added features like the ability to connect a siren and beacon and has output relays that can be triggered on events in the system. Commonly used protocols for sending and retrieving data is supported. A big plus for this system is that it has a built-in UPS which can provide power to the system for one hour in case of power loss. From the three systems studied, this is the only one that has this. A suggestion for how NTIs system could be used to monitor the laboratory can be found in figure 1 in the appendices on page 114.

Conclusion

In the end we chose not to use this system for several reasons. The device it self has 8 dry contact ports, 16 RJ45 ports and we need equal amounts of both. This means that for every 4 racks in the laboratory we need one EMS16U if all the requirements are to be filled. This also means that for every EMS16U there will be 8 RJ45 ports that will not be used which is

a waste. Compared to the cheapest alternative the total price to meet the requirements are unreasonably higher. The system is less modular than the other alternatives which means we are paying for options which might never be used. Also since we have to max out the number of units under one centralized server at once, the only way to expand is either by adding a second monitoring network or buying NTIs Server Environment Monitoring System Management Software which can monitor up to 3000 EMS16U devices at once depending on which package is bought[15]. If the laboratory at a later time would like to expand, the team would not be able to keep this system small and easy to understand for newcomers with this system.

3.2.2 Jacarta

Jacarta, based in Marlborough, England, is a company which specializes in environment monitoring equipment. Their interSeptor series has a set of devices called iMeter designed for environmental monitoring in data centers. The concept of the iMeter is the same as with NTIs EMS16U but with a more modular approach. With this system all the requirements can be achieved.

Figure 3.2: *Picture of iMeter Master. Property of Esis.com*

The iMeter package is made up of a master node, daisy chain-able slaves and the Go-Probe sensor accessories that can be connected to any master/slave node to get a wide array of different types of environment measurements and states. Up to 72 slave devices can be connected to the masters 4 extension ports (18 daisy-chained slaves per extension port) making the system able to provide monitoring for up to 576 directly connected Go-Probe devices. The whole system can be maintained from one single place and the user interface is clear and easy to learn for any person with basic computer skills.

The master and slave nodes does not have built-in ports for dry contacts. The documentation found on Jacartas home page is not very clear on this so communication with Jacarta was established to get a proper idea of what is

possible and not with this system. Unfortunately the correspondence between us did not make our understanding of the whole system any better.

A suggestion for how to use the Jacarta iMeter system to provide for the laboratory's needs can be found in figure 2 in the appendices on page 115.

Conclusion

In the end this system was not chosen either for several reasons, most of them not directly related to hardware, but instead the supplier and its documentation of the products. It was difficult to get a clear idea of what is available by using this system and when approaching the company with questions the situation did not get better. In the end the choice of not using this system was taken.

3.2.3 AKCP

AKCess Pro (AKCP) is a company with its headquarters in Hongkong with retailers all over the world. They manufacture and sell amongst other things networked environment monitoring tools. One of their products, the securityProbe 5ES seems to be a near copy of Jacartas EMS16U.

Figure 3.3: *Picture of SP5ES. Property of Ravica Blog*

As with the EMS16U this system is also built around a modular system with one master and daisy-chained slaves with the Intelligent Sensors series providing environment data to the system. Doing it this way with the Intelligent Sensors the system can handle up to 501 sensors at the same time which is more than needed by the lab now and probably in the future too. Since the

system uses slave units to expand the system, they can be invested in when needed and as they do not have the same capability as the master unit, the price of each unit goes down drastically, reducing the total investment needed to monitor the laboratory.

The system is built around the master unit running Linux with a 2.6.35 kernel compiled for Advanced RISC Machine (ARM) architecture with 128MiB Random Access Memory (RAM). The unit has been designed to be configured and maintained using its built-in web interface, which is easy to understand, easy to manage and most options are reachable in few clicks. It is also possible to connect to the system using Telnet and Secure SHell (SSH) (both turned on by default) and do some management from the terminal, however this is not documented well.

The systems base units does not have built-in dry contacts, but it can be installed by adding an expansion board called IO-Digital8 which can be plugged into one of the nodes 8 RJ45 ports, extending the node with 8 dry contact ports. The IO-Digital8 is able to work in input and output mode. When working in output mode, each port on the IO-Digital8 can sink 20mA. This is less than what is needed to drive the circuit breaker in the distribution panel to get the rack on/off functionality discussed in 2.3.3. So a solution would have to be developed to get this working.

AKCPs Intelligent Sensors series has all the required sensors to give us the ability to monitor humidity, ambient air temperature, rack temperature, egress water temperature and water leakage. A suggestion for how to monitor the laboratory using AKCPs equipment can be seen in figure 3 on page 116. Using this suggestion to monitor the laboratory, a reasonable amount of ports will be occupied yet still leave some ports on each slave unit available for expansion or impromptu use in different situations.

List of products needed to be bought to monitor the whole lab

Prices registered on the 02 October 2012

Quantity	Product code	Purpose	Price total
1x	SEC5ES	Master node for the system	$995.00
6x	E-Sensor8	Expansion boards (Slaves)	$1800.00
6x (7x)	IODC8	Dry contact adapter	$390 ($455)
1x	RWS10	Water leakage sensor	$250.00
5x	RWSCAB10	10ft. Ropewater extension cable	$500.00
1x	THS00	Temperature and humidity sensor	$120.00
1x	SS60	60 Feet door security sensor	$135.00
27x	TMP01	Temperature sensor	$2025.00
3x	DCT00-8	Daisy-chain temp pack	$1500.00
		TOTAL	$7715.00 ($7780.00)

The intension is to use the daisy chained sensors (DCT00) to measure in-rack temperatures and the dry contact expansion board to provide information about the doors and turbines in those doors. This will require 2 of 8 RJ45 ports for every four racks. 4 cable mounted in-line temperature sensors (TMP01) will be used to measure egress water temperature which will occupy 4 more ports in each local node. In the end 2 RJ45 ports per node will be left for future use where one will be reserved for expanding each node with 8 more dry contact ports to control power in each rack.

Redundancy and Reliability

The securityProbe 5ES does not have redundancy in terms of power supply. Because of this there are some reliability and security issues that must be made aware of.

Since it is not possible to connect the system to two or more power inputs, there is a real possibility that the whole or parts of the environment monitoring system will go down if the power supply on a master or slave unit break down. To alleviate this somewhat, a UPS can be used and also a power rail can be set up in the racks so that the units does not have an external power supply but get power directly from this power rail instead.

If the functionality to turn on and off power in individual racks is to be implemented, a standpoint on what to do with power in the racks has to be taken if the unit that is monitoring those racks dies. An argument is that the power in those racks being monitored by that unit should be shut down as there is no longer any way to get information about the environment in those racks. The way this would be implemented is by choosing relays that are open by default and need power to hold the circuit breaker in each rack down. If the AKCP unit dies, the rack loses power too.

Conclusion

In the end we chose the system from AKCP to provide us with the monitoring services. Their product fits the laboratory needs very well. It is light-weight, easy to understand, provides a wide array of accessories to monitor the laboratory with. It is scalable and the end price for the whole system will be less than for the two alternatives. In addition the documentation provided by AKCP is much better than the alternatives and all communication with them

went very smoothly. The first batch of devices to be used in the laboratory has been delivered and no major issues has been identified.

3.3 Deployment

3.3.1 Calibrating Sensors

Before any of this equipment can be put into production it needs to be tested thoroughly for expected behavior and deviation from the specifications. Several tests has been done on the system and sensors to identify possible problems and non linearity in the range of interest.

The range of interest for water will be from ~16°C (nominal ingress water temperature) and up to ~18°C (nominal egress water temperature). Since this system does not have a very fine resolution (0.5°C) it is important that the sensors are in fact representing the real temperature of the water running inside the tubes within its accuracy specifications. This is important because the system has less opportunities to discover that the temperature of the water is changing to unwanted/dangerous levels, inaccuracy here will make it more dificult to judge early warning signs for water cooling specific problems and it gives less opportunities to fine tune water supply to each rack.

For air, the range of interest is from ~10°C and up to ~50°C. Although the temperature span is much larger, precision is a requirement here too. It is also very important to know that the temperature output is linear over the field of interest. This is important because decisions about shutting down equipment will be taken at 30°C to 35°C, if the output is not linear the claimed temperature will not be correct and damage to equipment might occur.

For the purpose of calibrating and comparing sensors, a infrared thermometer and a high precision thermometer was invested in. The precision thermometer can measure temperatures from -10°C and up to $+60$°C. The temperature measurements will be compared to the thermometer and the infrared thermometer will be used as a control.

Temperature Sensors

Figure 3.4: *Temperature Sensors*

(a) TMP01 - Property of AKCP.com

(b) TMP00/DCT00 - Property of AKCP.com

Three different temperature sensors has been bought for the laboratory. The TMP00 which is a single unit air temperature sensor, the DCT00 which is a daisy-chain-able version of the TMP00 and the TMP01 which is a cable mounted inline sensor.

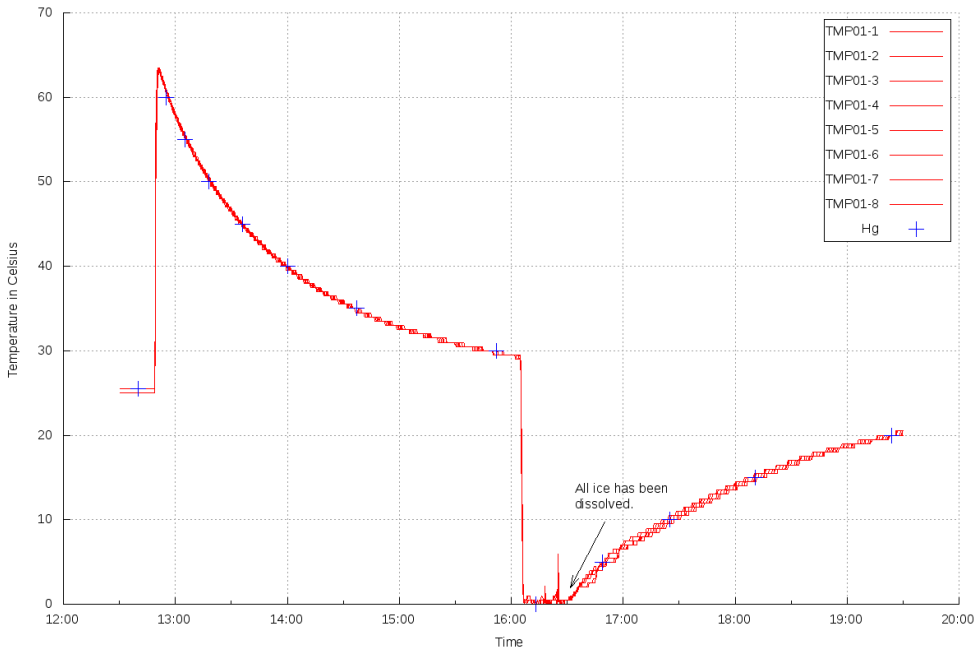
All three types of temperature sensors can measure temperatures from $-55°C$ and up to $+75°C$, has a 0.5°C resolution when using the securityProbe 5ES and has an accuracy of ±0.5°C in the range −10°C and +75°C [1][2]. This means that the sensors are able to handle the temperature range we expect in the laboratory.

TMP01

To calibrate the TMP01 a calorimeter was first filled with water and heated up to above $60°C$. When $60°C$ had been reached the precision thermometer was put into the warm water together with 8 TMP01 which had been put into a plastic bag prior to the test. The calorimeter and sensors were then left to cool down to room temperature. It became obvious almost at once that the bag did not give the sensors proper heat transfer as some of the sensors were off by several degrees C all the way through the test.

A second try was done where all the sensors were individually wrapped tightly in plastic and then the same procedure was repeated. This made a much better thermal connection and all the sensors converged to a common temperature. When the warm water reached room temperature, ice was added until the liquid reached 0 degrees C and the process of bringing the liquid to room temperature was repeated. It is very clear when viewing the graph in figure 3.5 that the TMP01 tracks the temperature very well. When subtracting the

Figure 3.5: *8 TMP01 tested for convergence*



individual sensor value from the mean ref. figure 3.6 it becomes obvious that drastic temperature changes will push the sensors to report values outside of its $\pm 0.5°C$ precision specifications.

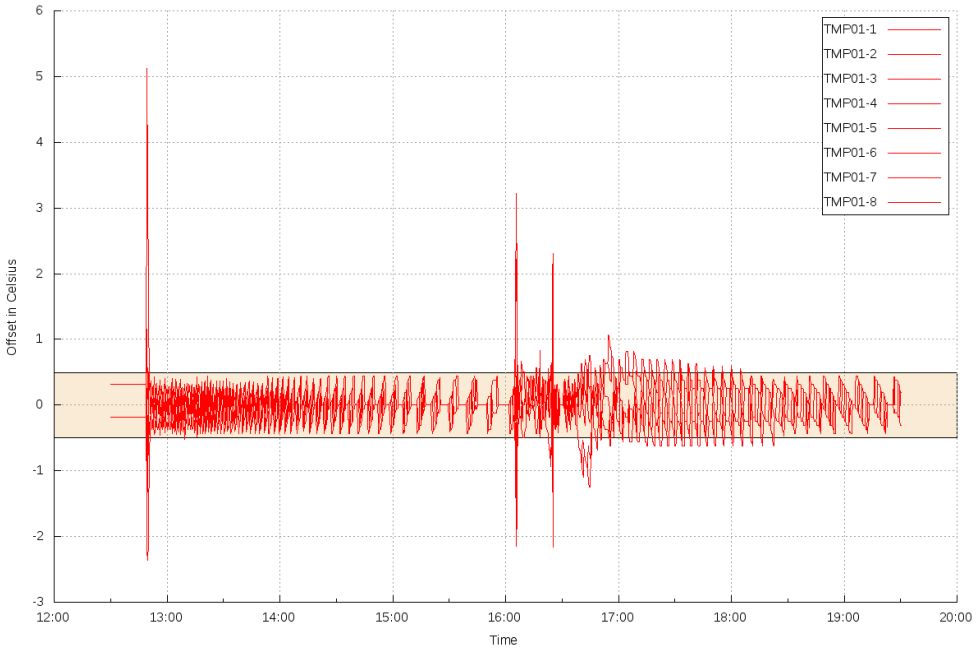
When cooling down the warm water and heating up the cold water the temperature of the sensors were compared to the thermometer every 5 degrees C. The resulting numbers can be seen in the appendices, figure .4.2 on page 117.

TMP00 and DCT00

Since these sensors are designed to measure air temperature, the previous method to test the sensors can not be used. So to test these sensors, 8 x DCT00 and 2 x TMP01 was strapped together, put on a table top and left there for a longer period of time to see if they would converge to a common value. The test showed that they would all converge within their accuracy rating at $20°C$ to $25°C$.

Extending the test range however was much more difficult than anticipated. Putting the cluster of sensors onto an open oven-shelf and into the oven did not work. The sensors would have a wide temperature spread and would not

Figure 3.6: *Temperature offset from the mean. Anything within 0.5 degrees Celsius from the mean is acceptable.*



converge. After some investigation there were three effects causing this.

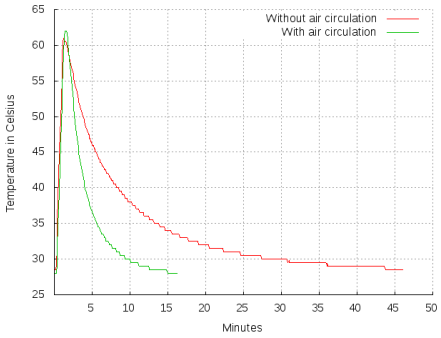
1. The oven has a really poor temperature control.
2. There is a really big difference in response time of the TMP01 and DCT00/TMP00 due to:
   - The limited ventilation of the TMP00 and DCT00.
   - The thermal mass.
3. Uneven heat distribution in space caused by draughts and uneven heating of the surrounding surfaces.

The latency is illustrated in figure 3.7a and 3.7b and is taken from some tests where each sensor type was heated up using a hairdryer and then left to cool down with and without a generic room fan blowing directly on it to provide ambient airflow. The graphs are clearly showing that there is a large difference in latency between the sensor types and that both prefer passing ambient air flow.

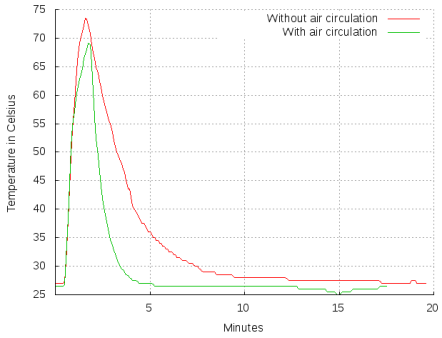
In fact, without ambient air passing the sensors, the time it takes to return to ambient air temperature is more than three times as long for all three types

Figure 3.7: *Tests done to discover latency issues in sensors*

(a) Comparison of TMP00 after heating it above 60 degrees Celsius



(b) Comparison of TMP01 after heating it above 60 degrees Celsius



of sensors. This explains the failure to converge in the first oven, probably the oven warms up and cools down faster than the TMP00 and DCT00 can respond.

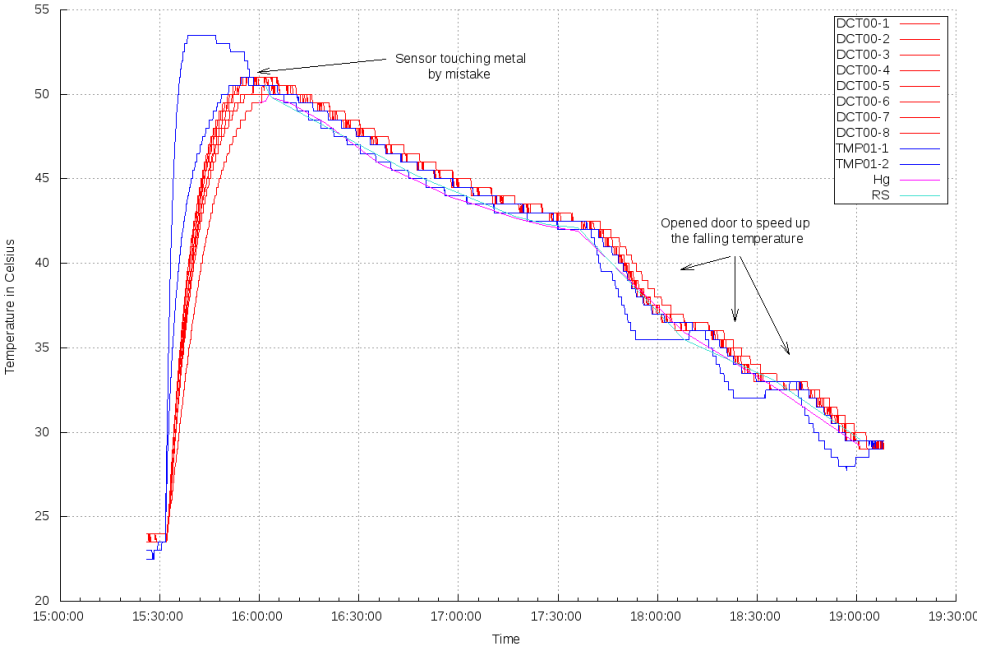
A second test was done with a different oven where a fully enclosed heavy metal pot was preheated to a stable temperature of ~60°C for an hour to make sure that the whole volume and surfaces were at the same temperature. The sensors were then placed in the pot and from that moment no further heating was added, only the oven fan was kept on to keep the whole volume at an even temperature. The rising temperatures of the DCT00 seen in figure 3.8 met the slowly falling temperature of the pot at 51°C at 15:54. From this point on TMP01 tracks the falling temperature nicely but DCT00 lagged slightly behind until the cooling rate slowed down at 46.5°C. At this point the gradient is close to 1°C/9$minutes$ 34 $seconds$ on average which is the response time for these sensors in the absence of air flow. When comparing the offset from mean figures for DCT00 (3.8) and TMP01 (3.6) there is clear proof that the TMP01 is much faster to converge back to ±0.5°C from the mean.

Conclusion

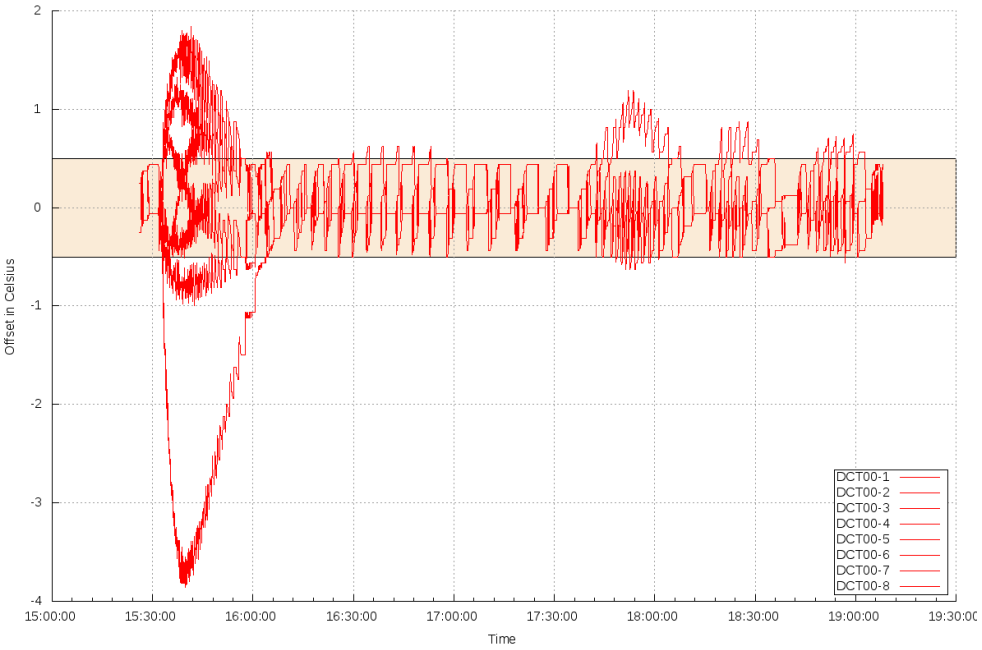
The conclusion from testing the sensors and studying the results are that all the temperature sensors are heavily dependent on strong air flow for early restoration of the real temperature after a heating episode. The sensors inertia is much larger for the TMP00 and DCT00 which partially is caused by the lack of local ventilation and the thermal coefficient of all the directly

Figure 3.8: *Graphs depicting the convergence test results.*

(a) Temperature convergence test



(b) Convergence test offset from mean



connected materials. Because of the poor local ventilation the sensor retains heat from earlier higher temperatures, looses it with difficulty and keeps affecting the sensor for a period after the incident.

For detecting rapid rises inside the room or rack all the sensors meet the requirements but the values must be discounted for several minutes after a rapid rise in temperature. For the ambient air measurements which are made in places outside of the racks with very little air circulation the readings from TMP00 and DCT00 must be discounted or the TMP01 can be used instead with advantage because of the lower mass and faster response times.

The tests done also show very clearly that all three types of sensors will converge within their precision specifications given enough time to do so. In conclusion we can use the DCT00 and TMP00 for measurements in significant air flow. For contact with the water pipes and for ambient air measurement in low flow conditions we will use the TMP01.

3.3.2 Placement of Sensors

TMP01

The TMP01 is being used in two different types of places, one being on the egress water hoses in the racks and the other is on strategical places on the ingress and egress tubes for the water supply in the laboratory.

In each rack a TMP01 is fastened to the egress water hoses by covering the sensor in a thick insulation material and then strapped down using vinyl ties. This provides good thermal contact between the sensor and the hose itself and reduces the influence the rack ambient has on the temperature readings to a minimum.

The sensors placed to monitor the ingress and egress water for the whole laboratory will be placed on tubes that is already insulated so a hole was made in the insulation of each tube and a sensor was pushed in and under the insulation on each tube. Velcro was used to hold the sensors in place.

DCT00

The daisy chain-able sensors has been fitted in the back of each rack in the middle of each grill on each heat-exchanger. This will ensure air rushing past

it, negating most of the negative effects of the limited ventilation discovered in 3.3.1 when calibrating the sensors.

While deploying the sensors the DCT00 were placed far down on the grill. This had the effect that all the sensors were showing unnaturally low temperatures. After some investigating, it was found that because most of the used racks in the laboratory are covered with blanking panels, some or most of the air being pulled through the turbines in the lower part of the racks come from the hole in the bottom of each rack because there is less restriction of air flow from there.

Two things had to be done to fix this problem.

- Move the DCT00 up higher on the grill.
- Cover up the hole in the bottom as good as possible with plates and cable brushes.

While deploying the DCT00 chains it was discovered that the ID given to each sensor seems to be random in some cases. When setting up for example a chain of 4 DCT00 and the sensors were given the IDs 92,93,94 and 95 it would not always be the case that the first and last sensor in the chain would have the IDs 92 and 95.

IO-Digital8

The IO-Digital8 has a shape and fastening mechanism that made it difficult to find a good place to fasten it inside the racks of the laboratory. It was in the end fastened with double sided tape on top of the slave units together with the power supply. Each heat-exchanger door-switch and turbine connection was then connected to the IO-Digital8 using signal cable.

Water leakage sensor

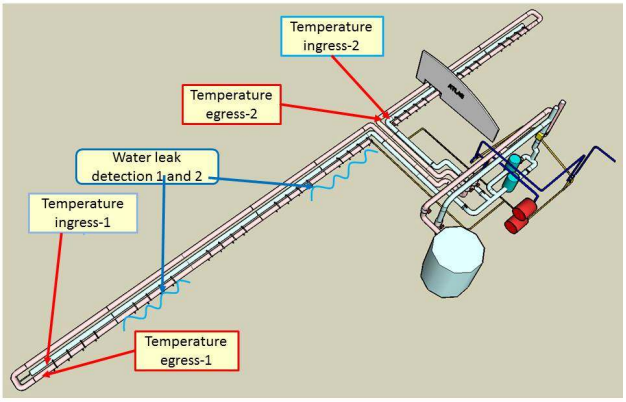
The water leakage sensor will be fitted on the floor in the plenum under the racks using double sided tape and brackets. The placement was chosen for two reasons.

- The heat-exchangers is the place where the warm, and potentially moist, air from the computers meets the cool metal of the exchanger. If the temperature of the metal is lower than the ambient dew point

then this is where the condensation will first form and drip onto the sensors below.

- The doors on each rack are constantly being opened and closed which creates wear and tear on joints. If the joints that connect the heat-exchanger to the ingress and egress water supply breaks, there will be a lot of water on the floor in that area.

Figure 3.9: *Overview of the position of each water leakage, ingress and egress water sensor[14]*



Humidity and Exhaust Sensors

There is one humidity sensor installed in the middle of the lab on one of the roof beams and two TMP00 installed on the back wall, one on each end of the room.

The humidity sensor will be used together with one of the two ingress water supply sensors to calculate ambient dew-point and differential ambient dew-point.

The two TMP00 installed in the back of the laboratory will monitor the exhaust temperatures from the racks in the general vicinity of that sensor.

3.3.3 The Sensor Network

The AKCP master unit has been placed in rack TD03 and the slave units has been placed in rack 1, 5, 9 and 13. All of the units occupy U 47 which is

the top slot in each rack.

AKCP Master Unit

The master unit functions as a star point for the currently four slave units installed in the laboratory. The slaves are connected to the master unit in such a way that port E1 has two slave units (TD01 and TD05) daisy-chained and port E2 (TD09) and E3 (TD13) has one slave unit connected. This will leave port E4 open for two slave units daisy-chained when rack 17-24 will be put into use.

The 8 sensor ports on this unit are being used for miscellaneous sensors like water detection sensors and similar which are not a part of the "standard" rack setup.

AKCP Slave Units

The slave units have a "standard" port setup to monitor the island the slave unit has been installed in.

Figure 3.10: *Standard port setup for a slave unit*

- Port 1: Used to connect 1 IO-Digital8.
  - Door sensors - port 1-4
  - Fan sensors - port 5-8
- Port 2: Used to connect 4 daisy-chain-able DCT00 .
- Port 3-6: Used to connect TMP01.
- Port 7-8: Not used, for later expansion.

3.3.4 Testing the Deployed System

The deployed system was tested by heating each DCT00 and one TMP01 up and above the preset warning levels in Em-poller ref. chapter 4. The water sensors were tested by pouring water on them. A random set of sensors were pulled out to test if Em-pollers function to report dead sensors were functioning.

All the sensors tested and Em-poller responded as expected and emails containing error and recovery messages were received.

3.4 Power Control and Autonomous Security

3.4.1 Power Control

In 2.3.3 it was mentioned that it is possible to turn on and off the power in individual racks using a PLC. To do this, each rack will need one port on a IO-Digital8 set as a output port and set to high when power is wanted and low when the power should be turned off.

There are some difficulties needed to be solved before this can be done. The IO-Digital8 is not capable of providing high enough current to the system so that the contactor keeping a rack on can be kept in place. So a device would need to be placed in between to make this work. Also since we do not have UPS for the environment monitoring system, we can not guarantee that they will stay on in case of power loss and if one of the environment monitoring units go down, all the racks connected to IO-Digital8s on that environment monitoring unit will also lose power. This could potentially turn the whole lab off because of power failure in one rack depending on device placement. There are many variables that can go wrong here so the decision was made to study this further before any implementation will be done.

3.4.2 Autonomous Security

The air volume compared to the number of devices radiating heat into the laboratory can loosely be compared to SDX and it was shown in 2.2.6 that it is possible to get temperatures above what the electronics have been designed to handle without breaking down in a very short time.

For this reason it would be favorable if computers in the laboratory could take a decision or take a command from a external source about when to shut itself down in case of problems which are escalating at a high rate without anyone watching or knowing about the situation.

The securityProbe 5ES is capable of triggering scripts at predefined temperatures and situations or use SSH to log into machines registered in a local database and shut them down automatically. This requires someone to keep the local database up to date and is a potential security risk since the device would have to be able to log in as a user with high enough privileges using one common Digital Signature Algorithm (DSA) Public Key Infrastructure (PKI) key to turn off one or more computers.

The Sys-admins has developed scripts and infrastructure around these scripts to shutdown multiple machines in parallel in case of another situation like the one mentioned in 2.2.6. These scripts can not be used directly but initial discussions mentioned it might be able to use the infrastructure around these scripts to get this to work in the laboratory too.

Chapter 4

Environmental Monitoring Software

A hardware solution has been found to provide the laboratory with environment monitoring. To use this available data a set of services need to be set up or created around it so that the data can be presented in a way which the users of the laboratory can understand and use.

4.1 Visualization tools

The laboratory has a Apache HTTP service running on one of its three network monitoring servers. It provides Net-IS and RackMonkey for the users of the laboratory. Net-IS together with a set of statistical data polling engines makes the Net-admins able to monitor network performance and congestion with little delay in time plus it provides historical data for later analyzing. In SDX, Net-IS is also used to monitor the environment through DCS ref. 2.1.5 but DCS can not be used in the laboratory hence this and the previous chapter.

The securityProbe 5ES comes fully prepared with a graphical interface for managing the device, sensors connected to it and a set of additional services which can be easily accessed by pointing any web browser to the securityProbe 5ESs IP address. One of these services is a Nagios daemon which will graph all the data it has collected in its local Round Robin Database (RRD) files. This is convenient if there is only one service of this type but in the laboratory Net-IS is already in use making the securityProbe 5ES solu-

tion redundant and the lesser choice because it introduces one more tool to learn, one more place to go for information and one more tool someone has to maintain.

4.1.1 Net-IS

For the reasons mentioned in 4.1 Net-IS is the preferred place to go for all information related to network statistics and environment data. To plot the graphs found in Net-IS the data contained in the securityProbe 5ES must be retrieved and stored in RRD files where Net-ISs graphing tool can access it. The existing solution to do this in SDX can not be used directly without further development which means a solution for this has to be found.

Since Net-IS was designed for SDX and its equipment the software has no concept of how to read data from the RRDs or what kind of data to expect from these RRDs in the laboratory. For this reason Scientific Linux CERN (SLC) 6 was installed on a virtual machine using Virtualbox, then Apache HTTP and Net-IS was installed on top of this again and modified so that it could corelate specific graphs in Net-IS to a RRD data source.

4.2 Research for existing tools

As mentioned in 4.1, a set of services need to be found so that the data contained in the securityProbe 5ES can be retrieved using Simple Network Management Protocol (SNMP), analyzed and stored in RRD files where they are reachable by Net-ISs graphing tool.

The securityProbe 5ES uses RRD files to store its sensor input. Since the unit has a Linux OS it was an early idea to expose the RRD files so that Net-IS could read them directly from the securityProbe 5ES. This idea was dropped for several reasons.

- The securityProbe 5ES might not have the extra CPU time and RAM to handle the added load with satisfaction.
- There is no documentation on how the securityProbe 5ES stores data in its RRDs which would make it difficult to do the necessary changes to Net-IS.
- All control over resolution, length of time to save data, how aggregation is done in the RRDs etc. is lost.

A search for possible suitable software packages were done and four interesting software packages were chosen for a closer look. Apoll, which is used in SDX today, Apoll Next Generation, Cacti/Spine, Collectd.

Apoll, which is used in SDX, can not be used together with the securityProbe 5ES without significant further development. Apoll NG which addresses many of the problems with Apoll is still under development and is not ready for deployment. Cacti with Spine is functional but we only need the Spine part of the package which makes it bloated for our needs.

Collectd seemed especially interesting as it provides a way to poll information using SNMP, store the received data as RRD files using one software package and not have any graphical user interface on top of this. However, it was difficult to get it to work as expected. In the end all of the choices were disqualified. None of the software packages and solutions looked at were satisfying or worked as expected.

A discussion was started to see if creating our own poller could be beneficial. The benefits of being able to decide ourself which properties and services the software package will provide out weighs the disadvantage of having to maintain the code ourself. Having spent some time studying the topic the decision was made to create a poller from scratch.

4.3 Em-poller

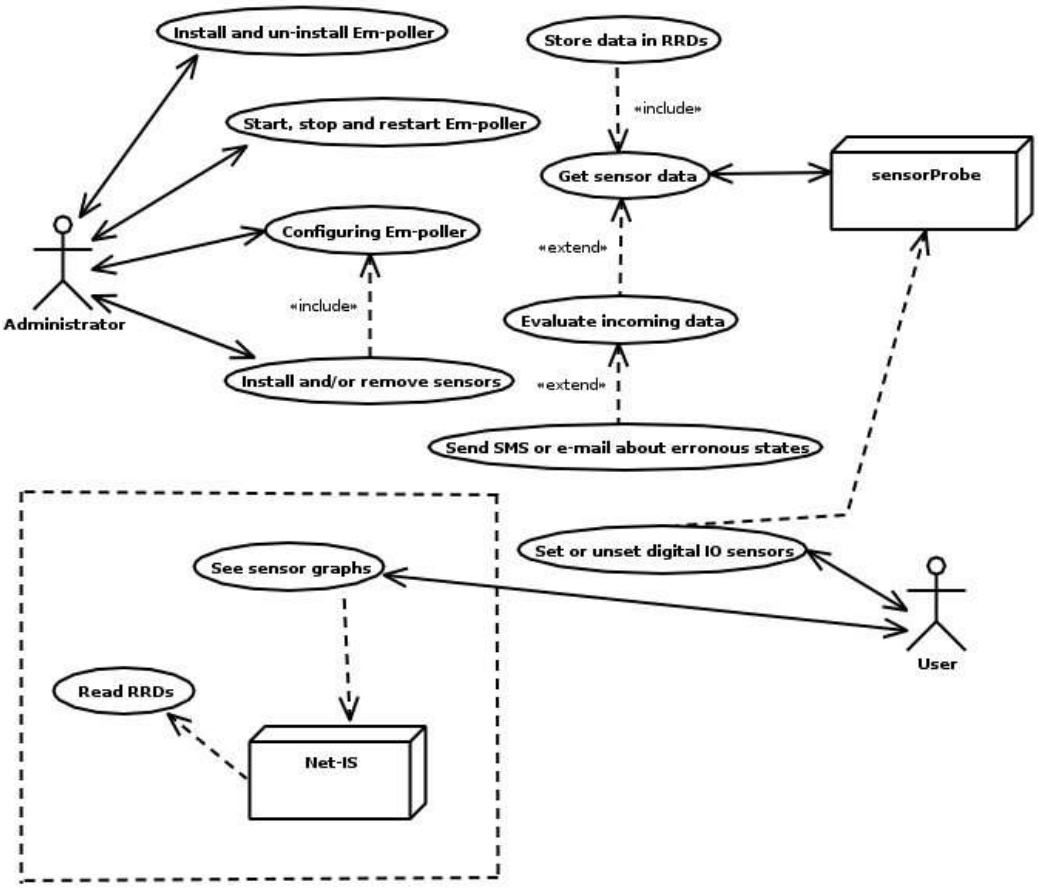
The software for getting environment data from the AKCP equipment will have to pull data from the securityProbe 5ES in a way supported (SNMP v{1,2,3}), analyze the data and store it as RRD files on a storage device where Net-IS can access it. In addition functionality like sending emails, warning of abnormal situations, logging and customizable configuration are much wanted properties of this service.

The Net-admin team are network administrators and not software developers. For this reason the software package should be easy to maintain and easy to understand. Preferably the interface to the service should be close to or identical to other services maintained by the Net-admins.

4.3.1 Use Cases

There are two main actors (administrator/user) and two secondary actors (securityProbe 5ES/Net-IS). Net-IS, although not a part of Em-poller, it has been included to show that it was necessary to add code to certain parts of it to interact with the system. Each use case will get a short description and a more detailed use case will be given for two of them.

Figure 4.1: *A use case model for the Em-poller engine*



Use case:	**Get sensor data and evaluate**
Goal:	Provide data for RRDs
Description:	The system will communicate with the securityProbe 5ES and get sensor data.
User:	The system will do this autonomously and only when erroneous state occur will any user be involved.

Use case:	**Start, stop and restart daemon**
Goal:	Start, stop and restart the daemon in a safe and orderly fashion.
Description:	Most Linux distributions have an initialization service which can start or stop any service automatically at boot or shutdown if specialized scripts are provided for the purpose. These scripts sometimes provide other mechanisms like restarting the service, reloading configurations and start the daemon in debugging mode. Em-poller will have an initialization script which will start, stop, restart and start debugging mode for the service.
User:	The administrator will run the "service" command with parameters to execute the scripts different functionalities.

Use case:	**See sensor data as graphs**
Goal:	Watch data accumulated over time from a specific sensors as graphs in Net-IS.
Description:	It should be possible to see each sensors accumulated data as graphs in Net-IS.
User:	The user will open a browser, point the browser to the server which Net-IS is hosted on and use the menus to guide him or her to the AKCP section and choose what type of information the user is interested in.

Use case:	**Install and Un-install software**
Goal:	Have Em-poller run as a service or removing it completely from the system..
Description:	An administrator should be able to easily recognize dependencies, install them before installing the daemon using a scripted installer. It should also be possible to remove the software using the same script.
User:	The administrator will run a provided script which does preparations, installation and removal.

Use case:	**Configure daemon**
Goal:	Configure the daemon to work in a specific environment.
Description:	The daemon must have a configuration file where all settings for the environment it is being used for is set. The service should be able to process secondary configuration files so that it is possible to use it in different setups.
User:	The administrator must create or edit and use the provided configuration file so that it fits the environment that the daemon is to be used in.

Use case:	**Set or un-set digital IO sensors**
Goal:	Tell the IO-Digital8 to set its output port to high or low depending on what result is wanted.
Description:	It is possible to control a IO-Digital8 so that it outputs logical low or high signals. This can be used by the system for many things like turning on strobe lights for error indication, enable relays to turn on and off racks and much more.
User:	Situation specific. Can be used manually by users or automatically by the system it self.

Use case:	**Install and/or remove sensors**
Goal:	Install and configure sensors so that they function correctly or remove not wanted sensors.
Description:	Install or remove a sensor and then configure Em-poller to use the new sensor or not use the removed sensor.
User:	The administrator will install or remove physical sensors, then grab its ID from the securityProbe 5ES interface and use that ID to configure it in the Em-poller configuration file.

Use case:	**Send SMS and/or email about erroneous states**
Goal:	Make a responsible user aware of erroneous situations that has occurred in the laboratory.
Description:	When a sensor of any kind goes into an erroneous state, a message (SMS/email) will be sent to what-ever email or telephone number set in the configuration.
User:	When receiving the message, the “on-call” user will have to respond to the message by finding the reason for the message and fix the problem.

Detailed Use Case Descriptions

Use case:	**Send SMS and/or email about erroneous states**
Goal:	Make a responsible user aware of erroneous situations that has occurred in the laboratory.
Description:	When out of normal situations like temperatures rise above preset warning levels, sensors disconnect from the securityProbe 5ES for what-ever reason or other incidents which cause loss of monitoring or danger to equipment a user which is "on-call" must be notified so that the issue can be rectified. Depending on severity level of the erroneous state an email or SMS will be sent to the user on shift.
User:	User on shift will have to investigate the problem and find a solution.
Type:	Essential
Conditions:	The hosting machines Mail Transfer Agent (MTA) must be configured correctly so that it forwards mails to the correct place. Em-poller has been configured correctly and email address plus telephone number has been defined in the configuration file.
Detailed sequence of events:	
1: System	The system will try to pull a sensor value from the securityProbe 5ES.
2:	If the system responds saying that a sensor is not available a warning message is prepared and sent. If the received value is a valid measurement, then the value is stored in a RRD and then evaluated by the system itself. If the received value is above any pre-defined warning levels, a warning email is sent to a pre-defined email address. If the temperature is above critical warning level then an SMS will be sent to the user on shift too.
3: User	User reads email or SMS, logs onto the CERN network and access Net-IS or securityProbe 5ES for further information. The user can now assess if it is necessary to get to the laboratory and fix the problem.
Alternative sequence of events	
	1 and 2 from the first sequence of events are executed.
3: System	The user for some unknown reason does not respond to the email or SMS sent. This will trigger the system to resend the warning email/SMS every **X** minutes where **X** is defined in the configuration file.

Use case:	**Install and Un-install software**
Goal:	Have Em-poller run as a service or removing it completely from the system.
Description:	An administrator should be able to easily recognize dependencies, install them before installing the daemon using a scripted installer. In the case that the software is to be removed the same script used to install the software should be able to completely remove the script or give clear instructions on how to do so. The procedure of installing or removing the software must be done in a safe and repeatable fashion.
User:	The administrator will run a provided script with parameters which does preparations, installation and removal.
Type:	Essential
Conditions:	The script will not install or remove dependencies so these have to be installed prior to installation of Em-poller. Root access is needed as the script will install or remove Python modules and create/delete directories and files in read only parts of the file system hierarchy.
Detailed sequence of events for installing:	
1: Administrator	Read the documentation.
2:	Install dependencies.
3:	Run the setup script with install parameters.
Detailed sequence of events for removal:	
1: Administrator	Take backup of configuration files and modified files.
2:	Run the setup script with removal parameters.
3:	Remove previous installed dependencies providing there is no other software installed which depend on them.

4.3.2 Programming Language

There are many choices of programming languages, some more fitting than others. Also there are many things to think of when choosing a language to write in.

- Speed of development.
- Ease of maintenance.
- Speed of the executing binary/script.

- Which languages are there good support and expertise for.

Because of the short time in which the monitoring must be developed, installed, tested, declared working and the fact that I will be working alone on this software limited the programming language options. Only languages I have some or much prior experience with will be looked at so that it is possible for me to increase my level of understanding and productivity in a relatively short time.

C++, a generic Object Oriented (OO) programming language which takes parts of high level and low level programming, combining them into a powerful tool was considered for its speed benefits, object oriented nature and wide support. However it was quickly decided not to code the software in this language as it is a bad fit in terms of development time and local support within the Net-admin team. Within months the software has to be developed, tested and installed in the laboratory making it essential that developing time is kept down which is not possible for a person without experience.

Perl, a high level multi-paradigm scripting language with wide support was also considered. Development time can be minimized by using the many libraries readily available using a Linux distributions package repository or Comprehensive Perl Archive Network (CPAN). Perl has a reputation of being very easy to write bad semantics in, meaning the code might be rendered very difficult to read. Although good coding practices will render this issue dead, it must be considered for later because there might be people with little or no experience in Perl that might work on the code. In the end this language was not chosen too because there is no local support for the language making maintenance a big issue.

Python, a high level multi-paradigm scripting language with a philosophy that focuses on code readability. Together with good code writing practices it becomes very easy to read most code snippets. Python has large support at CERN in general. CERNs IT department has courses in beginner, intermediate and expert level Python programming at regular intervals. This makes support and expertise easy to get. The standard library is big, making Python versatile and very fast to develop in which fits this project well. The speed of the executing code is slower than for the two alternatives since the code is compiled into bytecode and then run in a virtual machine. To make sure that Python could do the task some first page searches on google.com were done and it became clear that it is possible to get up to thousands of Object Identifiers (OIDs) per second if coded in a good way. This software needs to pull OIDs in the lower part of hundreds, not thousands so it is expected that Python will keep up with the demands. In the end Python

was chosen as the programming language for this project for its fast developing times, large support base at CERN and my previous knowledge and experience with the language.

4.3.3 Development Tools

All development for this specific project is being done with the following software:

- **Vim** has excellent Python support through add-ons which can be downloaded directly from distribution repositories or downloaded from the developers homepage and then manually installed. When configured properly and used as intended, Vim is a very effective tool for programming.
- **Git** was used for local revision control until the code was mature enough to be placed in the Net-IS Subversion project repository. The reason for using Git instead of Subversion as a intermediate were that Git is able to store local commits making it possible to keep close control over the source code without having to commit to the production Subversion repository with code that might not be good or stable enough to run on the network monitors in the laboratory.
- **Virtualbox** was used to run SLC 6 as a virtual machine on my desktop machine. This enabled me to test the code being written on the platform it would in the end be running on. In addition parts of the Net-IS source needed to be rewritten to fit in the changes that the laboratory environment monitoring brought. This meant installing a full copy of Net-IS, configure it for local use and change the code as needed.
- **Firefox** was used to debug Net-IS when issues with Django and Python arose.
- **Pyflakes**, **Pylint** and **Pep8** was used for code auditing and Python Enhancement Proposal (PEP)8 compliance checking.

4.3.4 Em-poller as a Service

The Net-admin team is not a systems administrator team or responsible for infrastructure other than the network itself. For this reason Em-poller should be as easy as possible to work with so that it does not increase the burden

on the team significantly. One way to do this is to make the service adhere to common practices within the Linux community.

SLC is based on Red Hat Enterprise Linux (RHEL) which uses RPM Package Manager (RPM) packages for software management. Packaging Em-poller as an RPM package would make installing and removing (purging) the service from the system easier and consistent. However since Em-poller will go hand in hand with Net-IS and all the other services and daemons used in the laboratory, Em-poller will not be served as an RPM package but instead committed to an existing laboratory branch in the Subversion repository for all the used tools in SDX and the laboratory. To make the installation and removal of the service repeatable and consistent, a Bash installer script will be made instead.

There is a standard called the Filesystem Hierarchy Standard[1] which explains in detail how the file system in any Linux should be. The standard states that any software that is to be installed locally and that should not be overwritten automatically when the system is updated should be installed in */usr/local*. In addition it states that any host specific configuration should be placed in */etc*[3]. Em-poller will adhere to both of these conventions.

SLC 5 and 6 uses SystemV for initializing and stopping services at boot, reboot or shutdown so initialization scripts that work with SystemV will be created and packaged together with Em-poller.

In addition a Cron script which will check if the service is running or not and email a preconfigured email address with information about the service not running when that is the case will also be provided. This script will reside in */etc/cron.d*.

4.3.5 Software Design

When designing Em-poller the idea has been to try and use as much of the standard library as possible and keep external dependencies to a minimum. In addition different logical parts of the program which is of generic nature will be pushed into its own files so that they can be installed as libraries at install time of Em-poller for added usability.

Pushing the generic parts of Em-poller out into its own files will reduce the amount of lines of code needed for the Em-poller in itself and enable others to reuse the written code for their project easier. The exported code will to

[1] Home of FHS

a large extent rely on parameter passing instead of importing specific non standard libraries so that each module can be used separately.

SNMP

The Python standard library does not have native SNMP support, so research was done to find different SNMP modules offered for Python.

- PySNMP: The only pure Python coded library.
- Net-SNMP: A suite of applications which provides Python bindings.
- YapSNMP: Yet Another Python SNMP module. Built on top of Net-SNMP.
- SNMPy: A library built on top of UCD-SNMP which is where the Net-SNMP library was spawned from. Not been updated since late 2009.

Net-SNMP was chosen from the various packages found, mostly because the package is required by a number of other services already running in the laboratory like Net-IS and the infrastructure built around Net-IS. By using this library we don't add dependencies to the infrastructure around Net-IS and we use software that there is already knowledge about within the team.

Threading

The first design of Em-poller was based on expectation of reasonable and consistent response times from the securityProbe 5ES. Once the first release of Em-poller was ready for testing it was found that polling data had breaks in it due to requests being blocked when the securityProbe 5ES was overloaded. This in turn would hinder Em-pollers ability to do its other tasks in a consistent manner.

To cope with this issue it was decided to rework Em-poller to use multi-threading so that essential services that Em-poller provide like error messaging and poll handling could still work as intended even though the threads pulling sensor data is in a blocking state.

When doing research on threading and Python a lot of information about the Global Interpreter Lock (GIL) come up. Some of these texts give a thorough explanation why threading in Python is a bad idea[9]. In the general case it is suggested to use multi-processing instead. Pythons multi-processing

module has been made to be almost transparently interchangeable with the threading module[10].

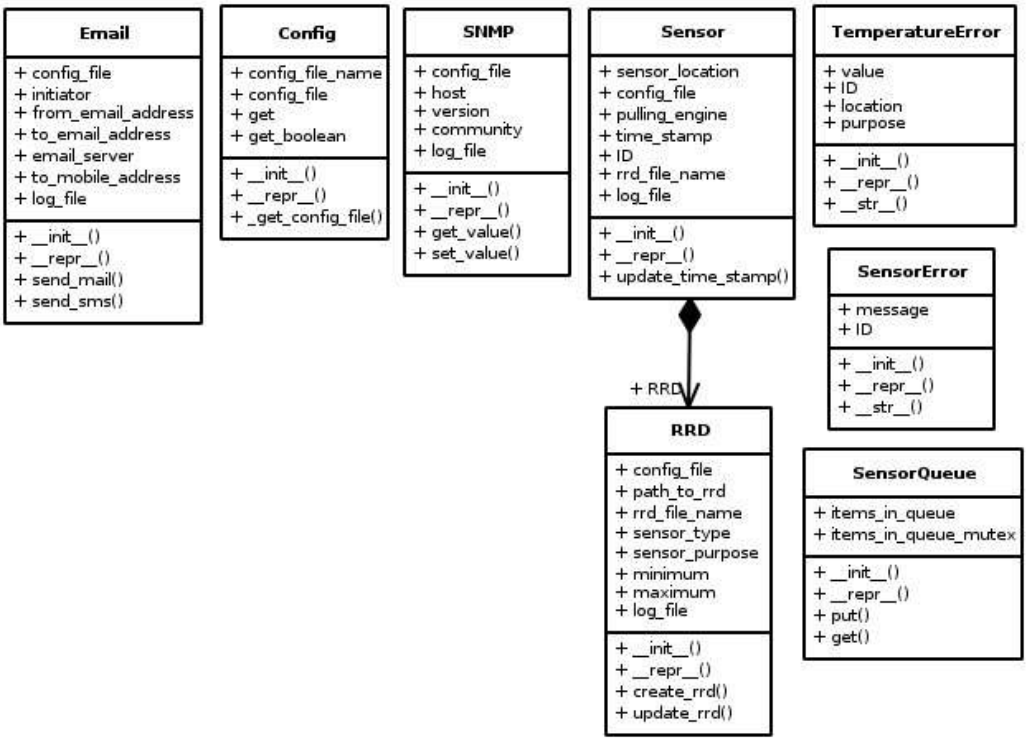
When looking at Pythons documentation on its own threading module[16] it says that if the spawned threads are expected to be Input/Ouput (I/O) bound then threading is still an appropriate model. This fits well with Em-poller since the intended threads will be spending most of their life either in a blocking state, waiting for SNMP data to be returned after a request has been sent or storing data in a RRD file.

Python Libraries

Em-poller in itself is not that many lines of code, in fact the specific code that makes Em-poller is about 500 lines of code and can be found in one executable in */usr/local/bin*. This executable however is dependent on 4 of the 5 modules created for the service.

- Config: A module that will provide a way for any Python code to read settings into their program.
- Email: A module that will send emails for any code provided that an Config object with settings for email addresses and server specifics is passed at creation time.
- RRD: A module that will create an RRD by getting RRD settings from a Config object passed at creation time. If it already exist, the RRD is loaded. Using built in methods the loaded RRD can be fed time series data.
- Sensor: A module that provides a way to read sensor values using SNMP and storing the received data in RRD files. The module provides helper functions to automatically return the correct object type by parsing a string passed to the function. The module also defines a set of Exception classes which is used by this module to raise errors that can be captured by code up the tree. Since the sensors respond in different ways, the module uses Polymorphism on common methods in sub classes to give a script or program a uniform interface to work with.
- SNMP: A module that takes a Config object at creation time and uses its settings to create an SNMP pulling object. This object can get or set any value for any OID passed to its methods.

Figure 4.2: *Libraries dis-jointed from Em-poller*



The files containing the modules are all named in lowercase, classes have names beginning with capital letters and the naming convention for any function, method or variable is all lowercase letters with underline between words as described in 4.3.7 on page 67.

Em-poller

Em-poller has been designed so that it is split into three parts. A control part, a worker part and a messaging part. Each part is running in its own thread/process except for the worker thread which can have as many spawned threads as needed and can be handled by the system.

When the software is started, the configuration file name and path is passed as a parameter by the init script to the Em-poller executable. The very first thing it will do is to fetch a copy of the configuration file and initialize all needed variables, settings and objects. In the case where the RRD files has not been created yet, they will be created at this point in the startup.

Control Loop

After initialization the process which spawns all threads in this service have only one task. That task is to iterate over a static list of sensor objects and check how long it has been since the object was updated with current information. If the time since last update for a sensor object is more than the RRD step time, then the object is pushed into a thread safe First In First Out (FIFO) queue.

While iterating over the static list of sensor objects the loop will calculate how long it has to sleep so that when it wakes up again there will be at least one sensor which needs to be updated.

Worker Threads

After initialization and the worker threads have been started the very first thing they will try to do is to pull a sensor object from the FIFO queue which the control loop pushes sensor objects into. Since there are no sensor objects in that queue yet, the queue will block all threads trying to pull an object from it until it starts receiving sensor objects which can be passed on to the waiting threads.

When a thread is given a sensor object, that thread will start the process of updating the sensor object's RRD file with current environmental data. If for any reason there is a problem, the sensor module will raise one of three exception types.

- SensorError: A generic Exception sub class which describes any physical problems that the securityProbe 5ES and a sensor may have like for example that a sensor is not reachable from the securityProbe 5ES.
- TemperatureError: An Exception sub class which has three sub classes, InfoTemperatureError, WarningTemperatureError and CriticalTemperatureError. One of the three are raised when a sensor temperature rises above the temperature defined for that warning level in the configuration file.
- LeakageError: An Exception sub class which will be triggered when a sensor reports that it is sensing water touching it.

If an error is raised and it is the first time that error has been raised for that sensor an email will be sent immediately, after the first error has been raised, emails will be sent when a sensor changes state. An example is when

a temperature sensor go from warning to critical state or vice versa. In addition an aggregated email will be sent for every time period configured as long as there is a sensor in any type of error state.

In the special case that a sensor changes state to Critical then an SMS will be sent immediately to the CERN mobile number defined in the configuration in addition to adding it to the list of errors to be parsed.

In the case that a sensor has become reachable after being flagged with SensorError an email will be sent with a message saying that the sensor has come back and the header "In-Reply-To" will be set to the identity of the first email warning about the sensor being flagged as in an error state.

Email Control Thread

The email control thread will check if there is any sensors currently in an error state by accessing a list of error messages and its belonging sensor IDs. If any is found it will parse the error messages and create an email containing detailed information about which sensors is currently in an error state, what temperatures those sensors has been seen having since the error state began and at what times those temperatures were seen. After sending the email to the address defined in the configuration file the thread will go to sleep for the time defined in the same configuration file.

When a sensor is no longer in an error state, the error messages and its belonging ID is removed from the list automatically without further due. The idea is to force a reaction from whoever is in charge of keeping an eye out for the system at the time of the event. This way a user will have to check what is going on and share the conclusion with the rest of the users receiving the email.

4.3.6 Software Testing and Auditing

To be able to test the developed code, a virtual machine was set up in Virtualbox with all the necessary dependencies and configured so that it would behave in a similar fashion as the network monitoring machines in the laboratory. The securityProbe 5ES was set up with one sensor of each type so that the system would be polling environmental data 24/7, tested and checked for unexpected behaviour at all times.

Source Code Auditing and PEP8 Compliance

At regular intervals the source code would be audited using the static analyzers Pylint, Pyflakes and Pep8. Each Python file was first passed through Pylint and Pyflakes to be checked for logical errors and then passed through Pep8 for PEP8 compliance.

Figure 4.3: *Example output from pep8*

```
hamartin@Lirael:~/em-poller/bin$ pep8 em-poller
em-poller:19:1: E302 expected 2 blank lines, found 1
em-poller:19:28: E261 at least two spaces before inline comment
em-poller:42:80: E501 line too long (80 characters)
em-poller:64:1: W291 trailing whitespace
em-poller:254:73: E202 whitespace before '}'
em-poller:474:31: E231 missing whitespace after ','
hamartin@Lirael:~/em-poller/bin$
```

4.3.7 Documentation

Em-poller will need documentation so that future users, administrators and developers can quickly read up on the needed information.

reStructuredText

The service itself should be well documented so that the administrators of the service have a place where they can look up general use of Em-poller. There is a set of utilities written in Python called Docutils which includes reStructuredText (rST). All the user documentation will be written in rST so it can be processed and read in many different document types including HyperText Markup Language (HTML), Portable Document Format (PDF) and "man-pages".

PEP 8 and PEP 257

The code itself will need documentation too and for this the two PEP documents 8[2] and 257[3] will be used extensively to provide source that is easy to read and contains the necessary information to give future developers the

[2]PEP 8: http://www.python.org/dev/peps/pep-0008/
[3]PEP 257: http://www.python.org/dev/peps/pep-0257/

opportunity to understand, add code or rewrite existing code with minimal effort.

Highlights from the two PEPs are:

- Any line in the source should not be longer than 79 characters unless breaking the line up will make the source code less readable.
- Choose one naming convention and stick to it.
- Use doc strings for documenting the different classes, methods and functions.

For Em-poller it was chosen to name modules with all lower letters, classes are named with starting capitol letter in each word (*CapitalizedWords*), variables, methods and functions are all lower case with underscore instead of spaces. In all of the mentioned cases underscore will represent space. Constants are either all upper case with underscore representing space or camelCase beginning with underscore.

Constants that have been named with underscore and camelCase will be constants that represent an AKCP Management Information Base (MIB) OID. The name will then be the exact same name as the tag and iid found in that MIB file.

4.3.8 Final Result

The polling interval has been set to two minutes per full scan. This has been shown to be error and blocking free for the full set of sensors and is bound by the performance of the CPU in the securityProbe 5ES. Technically there wasn't a "required" polling rate although my supervisor had indicated that 30 seconds was a reasonable goal. In practice this is not an issue if the thinking behind that goal is looked at more closely. The real concern is that cooling failures are detected as soon as possible to allow time to shut machines down and avoid equipment failure. The laboratory has the same or larger air volume and less power input than SDX so the SDX worst case figure of a 1 degree C per minute temperature rise will not be possible, but it serves as a useful upper bound. This is where the 30 seconds come from, being able to detect a one degree rise with two successive readings of one sensor. In the laboratory there are three ambient temperature sensors so three results are being returned every two minutes. This rate is close to match both the likely worst case of the laboratory and the measured response time of the sensors them selves so the requirements are considered to be met.

Em-poller has been running in the laboratory with no errors and no data drops for over three weeks. This included one brief and scheduled stop to update a configuration file after new sensors had been cabled in. An example of the output from the collected data stored in the RRDs is shown in figure 4.4 and 4.5.

Figure 4.4: *Example output from Net-IS - Laboratory view.*

Figure 4.5: *Example output from Net-IS - Rack view.*

Em-poller error messages

Figure 4 and 5 on page 119 show example emails received when a sensor can not be polled or the temperature of a sensor rises above one of the three defined error levels in the configuration.

Chapter 5

Evaluation & Self Assessment

At the beginning of my induction to the Net-admin team I was told that I would be given a major part of commissioning the Atlas laboratory in building 4 at CERN. I would amongst other tasks be given responsibility to port existing tools, reconfigure them and commission them for use ref. .1.1 on page 95.

The laboratory has been erected, most of the software has been ported and the facility has been in use for some time. Further work will be done by both the Net-admins and Sys-admins to improve its usability for the Atlas project.

5.1 Project Evaluation

5.1.1 Organization

The contact between me, Erik Hjelmås and Brian Martin has been satisfactory all the way through the project. In the beginning a bi weekly video conference meeting was held between me and Erik Hjelmås. Later in the year it was agreed upon by both of us that we would have these video conferences only when needed by one of the parties. This worked well.

The fact that Brian Martin was available for me when ever needed was a great motivational factor for me. We had contact on a daily basis and the continuous feedback he gave to me improved both my writing but also my understanding of the task at hand.

Because I had constant access to my supervisor while doing this project the use of Scrum and two week Sprints was abandoned early and replaced by constant communication between me and my supervisor. This worked much better because this gave both of us the option to resolve issues there and then instead of waiting for the bi weekly planning meeting to get the issue resolved.

5.1.2 Intended result/effect

All the intended results ref. .1.1 except for remote power on/off functionality has been implemented. The remote power functionality was not implemented because there are many issues around the implementation that has to be considered and discussed before it can be implemented.

- What should happen with the power in the racks a AKCP unit controls if the unit stop functioning?
- Is there knowledge within Atlas to maintain this in a safe and functional way?
- Since the IO-Digital8 are controlled using SNMP. How can these units be configured so that they only take commands from authorized users.
- Is the implementation safe?

Although it was decided not to implement this functionality at this time, the snmp module has been given methods to send commands to any SNMP supported device for use either by other software implementing this module or by Em-poller at a later time.

All the intended effects ref. .1.2 has been achieved and further work is being done to complement these results.

5.1.3 How Time was Spent

In the beginning of the project, most of my time was spent getting up to speed with the Net-admin teams responsibilities, getting a general feel for how SDX is built up and works. Then once the laboratory room was released from construction I contributed a lot to the initial setting up and testing of the technical infrastructure.

- Installing racks.

- Cabling, labelling and testing the network infrastructure.
- Preparing racks and equipment in the laboratory for future use.

When the racks and most of the infrastructure was in place the work with getting ready for installing the environment monitoring system began. This took significant time for several reasons.

- Net-IS is built on Django so I had to teach my self from scratch how Django works.
- Once I understood how Django work I had to read Net-ISs code for a longer period of time to get an understanding of how Net-IS works. Dan Savu at CERN was also kind enough to spend time to explain me how Net-IS work on code level.
- In SDX, Net-IS is running on SLC 5, in the laboratory it is running on SLC 6 instead to prepare Net-IS for future releases. This meant that dependencies changed and some code failed and needed to be repaired.

When sufficient knowledge about the tools and starting difficulties with installing OSs, needed daemons and software was resolved the work of getting a working environment monitoring system began.

Em-poller

Em-poller has been written, tested and deployed in the laboratory with great success. No major bugs has been discovered at this point in time[1].

5.1.4 Hardware

Working with the AKCP hardware was a good learning experience for the need to evaluate the parameters that are not explicit in company documentation. In particular not assuming that sensor specifications are correct and not making any assumptions about the performance of embedded processors.

All in all the AKCP system was a good choice even with the CPU limitations discovered after acquiring the system.

[1] 15^{th} December 2012.

5.1.5 Choice of Programming language

Choosing to use Python to program Em-poller in was a good choice. In hindsight, the arguments that Python is fast and easy to develop in has shown to be true. The extensive documentation for all its functionality made it very easy to learn and implement all the needed parts of Em-poller.

5.1.6 Issues for Later Thought and Further Development

The sensor module makes it easy to add different types of sensors that can be polled from. However, all the loaded sensor objects that are being polled must all be supported by the same SNMP server. This is because Em-poller is only able to poll from one host which is defined in the configuration file. For later development it would be a good idea to make Em-poller host aware and make it able to differentiate which hosts can be polled for what. This would make Em-poller a lot more flexible and could be used for storing time series data not only for sensors supported by securityProbe 5ES but also switches, routers and computers.

The emailing part of Em-poller can be improved significantly with refactoring. Late changes in the code broke the structured and clear border between the worker threads and email thread making it unclear which thread is responsible for what type of email messages being sent.

Because of the issues with the GIL and multi threading, Em-poller should be rewritten to use Pythons multiprocessing library or made so that the need for threads/processes are abstracted out.

When writing code in Python, it is common to use *args and **kwargs to specify arguments that a function or method should accept. This has been used in the code to a very little degree. Functions and methods should be re factored so that it implements this instead of specifying each argument that must be passed to the function/method.

5.1.7 Milestones

All milestones were completed on time and according to plans. On Thursday the 29^{th} November 2012 the laboratory was officially declared ready for use

to all contributors with a small drink in the laboratory and a speech from Brian Martin.

5.2 Self Assessment

My personal goals ref. .1.3 were to raise my technical paper writing standard, extend my knowledge and understanding about how computer centers work and how they are serviced in general.

My personal impression is that with the help of my supervisor and the Net-admin team I was able to heighten the quality of my writing enough to write this paper with satisfying result and get a general knowledge about computer centers which I feel will be a good asset to have later in life.

5.2.1 My Contribution

My initial training in the Net-admin team was to learn about the infrastructure within SDX and study Atlas literature describing the early requirements and first implementation of SDX. As I was studying SDX in detail it was natural to also study descriptions of data centers in the industrial and commercial fields in parallel.

While studying SDX in detail I also played a large role in dismantling a set of racks and their contents in SDX and then re-installing the racks and machines in their place as a part of a upgrade. With the experience acquired from this I went on to play a major role in establishing the new laboratory in building 4 and took responsibility for the installation, cabling and testing the switching infrastructure of the laboratory. When the initial networking infrastructure was in place I researched the options to replace the RackWizard functionality and was responsible for the final choice, its installation, modifications and deployment in the laboratory.

I worked on requirements for the environment monitoring together with my supervisor at CERN. The market survey and follow up of unclear technical issues for each product was my responsibility. The final choice of product was essentially mine as well. The definition of the languages, methodology, program design, implementation, debugging and documentation was all my own contribution. Help was obtained from the authors of previously installed packages to better understand their Application Programming Languages (APIs) but the actual integration of the packages was mine.

The initial attempts to calibrate and test the sensors were mine although help was sought from my supervisor when the initial results did not meet expectations. I contributed to the physical installation of the complete system, its initialization and testing was all my contribution.

5.2.2 Self Evaluation

The available time through the year has been spent well. There has been steady progression all through the year with a small peak towards the end of the project.

I feel that I learned to be flexible when facing practical hardware and software limitations. Since I was discovering both Pythons and the AKCP system functionalities and limitations while writing Em-poller, the architectural design of Em-poller changed a couple of times in the beginning of the development process.

Programming/Scripting in Python

When arriving at CERN my knowledge of Python was limited to the basics and any coding had to be done with extensive use of documentation. Now, one year after I have gotten more experience coding (scripting) in Python, I don't need to use the documentation all the time for the most common modules and built in functions.

The biggest changes for me other than getting better at using the standard library without documentation is Duck typing. In the beginning this felt strange and unnecessary. As time progressed the benefits of Duck typing became evident and has become one of my best arguments for using Python to code/script in.

Using Git for Version Control

Using Git for version control has been very good for the development of this document and Em-poller. Using Git made it possible for me to work from several different machines without having to worry about copying new revisions from one machine to another manually. It also enabled me to test out structure changes by creating temporary branches giving me the option to revert to older revisions if the changes did not work as wanted.

Using my own Git repository for the Em-poller source made it possible for me to commit changes frequently without spamming down the Net-IS Subversion repository with commit messages.

Typesetting in LaTeX

Choosing to use LaTeX for typesetting this document has been satisfactory but troublesome at times. Although well documented, it can be difficult to know what is possible and what is not when typesetting different parts of the document.

When getting past the initial troubles, LaTeX became a joy to write in and it gave me an appetite for writing and I will definitely use it more for future documents.

Glossary

A A derived unit of electric current (Ampere) in the international system of units (SI). . 9–11, 20, 37, 82

AKCP A company that provides environment monitoring equipment and services for industrial needs. . 36–38, 47, 53, 55, 68, 72, 73, 76, 84, 88

Alice One of the experiments being run at the LHC at CERN. Alice is an optimized detector for heavy ion collisions. . 22, 24

Apache HTTP A web server designed to serve media on and to the web. . 29, 51, 52

API An protocol intended to be used as an interface to a system by software components. . 75

Apoll A high speed multi-threaded SNMP poller developed in C by Dan Savu at CERN[17]. Its successor Apoll NG is being written in C++ . 52, 53

ARM A 32 bit CPU architecture running a reduced instruction set (RISC) from ARM holdings . 37

Atlas One of the experiments being performed on the LHC. . 1, 3, 5, 8, 14, 16, 18, 19, 24, 27, 31, 71, 72, 75, 85, 86, 88, 89, 95, 96, 100, 101

backbone Indicates a part of the network designed to switch and route packets at very high rates. This logical level of the network is not suppose to be taking decisions about if a packet is to be forwarded or not, but focus on transferring the packets as fast as possible to the next hop destination. . 27

Bash A widely used Unix
Linux shell and because the implementation is able to read input from a file it is also a scripting language. . 61

Bibtex A tool to process references in a Tex document. Home of Bibtex. . 101

BIOS A program installed on a chip inside the computer. This program is the very first thing to run on your computer when you boot it up. Its purpose is to check and prepare hardware for boot. . 18

blanking panel A panel that is installed in open slots on the front of the rack to prohibit air being pulled through areas in the rack that is not populated with hardware. . 5

broadcast address A logical address which all devices connected to a multiple-access network is listening to and can respond to. . 27

bytecode A set of instructions designed for efficient interpretation by a software interpreter like Pythons virtual machine. . 59

C A scale and unit of measurement for temperature. . 16–18, 32, 39–41, 43, 68

C++ A general purpose programming language. Developed by Bjarne Stroustrup in 1979. . 59

cable tie A cable tie, also known as a zip tie or tie-wrap, is a type of fastener, especially for binding several electronic cables or wires together and to organize cables and wires. . 6

Cacti A open source web based RRDTool front-end. . 52, 53, 88

calorimeter A device used to measure the heat loss or gain in a system. . 40

camelCase A way of describing a naming convention for variables, functions, methods and classes where the start of each word in the name is capitalized. . 68

CERN A research institution whose purpose is to do research on physics. The initials stood originally for Conseil Européen pour la Recherche Nucléaire. . 1, 21, 26–28, 56, 59, 66, 71, 73, 75, 76, 79, 81, 83, 84, 88, 95, 96, 99, 101, 102

chiller A machine that will chill a liquid by first compressing vapor to high levels and thus heating it up in the process. The compressed and heated vapor is then cooled and condensed into a liquid by letting it flow through a heat exchanger of some kind. The system will now abruptly force the high pressured liquid through an expansion valve,

making the liquid even colder and turning it into gas. The liquid will then be sent through the system to do its job before going back to the circle it started. . 15, 24

circuit breaker A thermo-magnetic device used for circuit protection which is divided into two parts. Both parts are connected serially, where the first part is a electro magnet which trips in milli seconds and the second part is often a bi-metallic strip which trips in seconds or more. The electro magnet works for power surges and the strip works for slowly rising currents in the system. . 10, 11, 20, 37, 38

Collectd A daemon that collects different types of statistics and can store them in a variety of ways. . 52, 53

compressor A device that will increase pressure by decreasing volume on a gas, forcing it to go into a phase change and become liquid. . 24

condenser A device that will condense a substance from its gaseous state and into a liquid. . 23

CPAN A place where Perl modules and libraries can be downloaded from. At the time of this writing it contains more than 114 000 modules. . 59

cPCI A computer bus for industrial computers. . 5

CPU A device inside the computer which executes instructions given to it by performing basic arithmetical, logical and input/output operations. . 8, 52, 68, 73

Cron A job scheduling daemon used for automating processes. Its general purpose nature means that it can be used for many different purposes like system maintenance and administration. . 61

daemon A process that runs in the background that is not under direct control of an interactive user but instead is meant to be left alone to do the task it has been given. . 51, 54, 55, 58, 61, 73, 81, 89

DCS DAQ system comprising the control of the sub-detectors and of the common infrastructure of the experiment and the communication with the services of CERN (cooling, ventilation, electricity distribution, safety etc.) and the LHC accelerator. . 10, 13–18, 21, 31, 51

DCT00 The exact same sensor as the TMP00, but with the added functionality that it can connect up to 8 of them to one port by daisy-chaining them. . 37, 40–43, 45, 46, 48, 89

DHCP A network protocol designed to configure devices so that they can talk to each other on a IP network. . 27, 88

differential breaker A device that disconnects a circuit when the device discovers that there is a difference between the amount of As coming out of the neutral line than what is provided into the circuit. Any difference is assumed to go to ground, perhaps via a person so the circuit is tripped within milliseconds. . 11

distribution A brand of Linux maintained by different people and companies. Some example of distributions are Red Hat, SLC, Debian, Ubuntu and Slackware. . 54

Django An open source web framework written in Python. The software is designed to ease creation of complex data base driven systems. It emphasizes making code which can be reused. . 60, 73

dry contact A type of switch that does not break or make a circuit but is used to relay information to other parts of the system. A contactor is often used for this purpose. . 32, 34, 35, 37, 84

DSA A standard for digital signature attributed to David W. Kravitz, a former NSA employee. . 49

Duck typing A style of dynamic typing where the objects methods and properties determine the valid semantics. In Duck typing one is concerned with objects properties and methods instead of being concerned about what type of object it is. . 76

Em-poller An SNMP polling engine which stores all collected data in RRD files. . 48, 53–56, 58, 60–64, 67, 68, 72–74, 76

EMS16U An environment monitoring system from NTI. . 33–36

e-reporting A generic term for a procedure that will collect data from devices or systems and report issues and status of those devices and systems. . 102

Ethernet A computer networking technology for local area networks. . 7, 34, 83, 87

evaporator A device that will transform a liquid into a gas through thermal contact with another media making it boil as it absorbs energy from the other media. . 24

FIFO Describes a queue where objects pushed into the queue leave the queue in the same order it was pushed into the queue. . 64, 65

Firefox An open source web browser coordinated by the Mozilla Foundation. . 60

GE Describes one Ethernet connection with a transfer rate of 1 Gigabit. . 27

GIL A function in programming languages that will prevent code sharing between threads in a process. This is done to prevent unsafe threading code to be executed simultaneously. . 62, 74

Git A version control tool to keep a strict control over the changes that happens in a project, coming from different users. Home of Git. . 60, 76, 101

GNU A project started by Richard Stallman. Its purpose was to provide an OS that was completely free. . 29

Go-Probe A set of sensors designed to work with the interSeptor line of devices sold by Jacarta. . 35

GPL A license used for general purposes. Any derived work from software licensed as GPL must be distributed under the same license terms. . 29, 85

GPN The network provided by the IT department at CERN which connects all general purpose buildings and places at CERN. . 17, 27, 28

green field project Means a project that does not have constraints imposed by prior work. . 14

HTML A markup language for displaying pages and other information which can be displayed within a web-browser. . 67

HVAC Refers to technology of indoor and automotive environmental comfort. . 4, 26

hypervisor A virtual machine manager. Allows multiple OSs to run concurently on a host computer. . 90

Hz Hertz, a derived unit of measurement of frequency. . 8

I/O Describes anything evolving data coming in and out. Often describes communication between hardware and software. . 63

iMeter A set of data center specific monitoring devices in the interSeptor series from Jacarta. . 35, 36

Intelligent Sensors A series of accessories to the base units AKCP sell. . 36, 37, 84

interSeptor A series of different types of environment monitoring devices from Jacarta designed for a wide array of different environments. . 35, 83

IO-Digital8 One of the products found in the Intelligent Sensors series sold by AKCP. The device can exchange 1 RJ45 port into 8 input or output digital ports (dry contact) on their base units. . 37, 46, 48, 49, 56, 72

IP A protocol which describes how devices can be given addresses on the Internet or on a local network. . 27, 28, 51, 81, 88

Jacarta A company that provide environment monitoring and equipment for industrial needs. . 35, 36, 83, 84

kernel A kernel is usually the main part of any OS. Its purpose is to act as an intermediate between applications and the actual hardware. . 37

LANDB A database used by CERNs many projects and departments to register machines so that they can be automatically given IP addresses depending on which service it is registered in. . 27, 28

LaTeX A document preparation system. Home of LaTeX. . 101

LHC An energy accelerator used for particle physics research. . 1, 8, 9, 79, 81

Linux A Unix like OS created by Linus Thorvalds and released in 1991. . 21, 32, 37, 52, 54, 59–61, 79, 82, 87

MIB A database used to manage entities in a communication network. Often referring to SNMP end nodes. . 68, 85

Moores law A rule of thumb in the history of computing hardware whereby the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years. . 8

MTA Software that implements the sending and receiving part of the Simple Mail Transfer Protocol (SMTP) protocol. As the name suggests it transfers mail from one place to another. . 56

multi-paradigm In context of programming; Describes a language which supports more than one programming paradigm like functional, object oriented, imperative procedural and more. . 59

MySQL An SQL based data base administration system licensed under GPL. . 28, 29

Nagios Open source software to monitor IT infrastructure. Widely supported and used to monitor all sorts of devices and useful information. . 51

Net-admin The network administrators is the team responsible for installing, maintaining and developing the Atlas DAQ network. . 27, 29, 31, 32, 51, 53, 59, 60, 71, 72, 75

Net-IS User interface for historical and statistical data gathered in the Atlas network. This both for network specific data and environmental data. . 51–53, 55, 56, 60–62, 73, 76, 101

Net-SNMP An open source software package providing SNMP applications and Python bindings to implement SNMP into a program. . 62

NTI A company that provides a large set of different types of services for industrial needs. . 33–35, 82

OID In SNMP context: An object identifier for an object in a MIB. . 59, 63, 68

OO Describes a way of programming where the data is in focus. Each data structure can have properties and can interact with other objects in different ways. . 59

Open Office Draw Open source software used to make sketches, plans and diagrams. Home of Open Office Draw . 101

OS A collection of software that manages your computer hardware resources and provide common services needed by computer programs. . 28, 32, 52, 73, 83–85, 90

OSI A standard describing the functions of a communications system. Similar functions are grouped into logical layers and works by the model starting at level 1 which is hardware and stops at level 7 which is software. . 87, 89

PDF A document format designed to be application, software, hardware and OS independent. . 67

PEP A document describing new features, design issues or processes to the Python community. . 60, 66–68, 86

Pep8 The eight PEP document and also a Python source code auditing tool that checks compliance with PEP8. . 60, 66

Perl A high level general purpose programming language developed by Larry Wall in 1987. . 29, 59, 81

pitch A distance between the same two points on similar objects. . 8

PKI A set of procedures, software and policies which is needed to manage use, store, revoke and distribute digital certificates. . 49

Planner A tool for project management. Used to create Gant schemes. Home of Planner. . 101

PLC A digital computer used for automation of electromechanical processes. . 10, 11, 20, 21, 48

plenum A space provided for air circulation and used to route conditioned air to where it is needed. . 3–6, 19, 32, 46

Point 1 The whole installation above and below ground that supports the Atlas experiment. . 17

Polymorphism A programming language feature where a function or method at a lower level can be "overwritten" by a higher level function or method so that different types of data can be worked on in a uniform way. . 63

product backlog A list of defined requirements ordered by the product owner [20]. . 98

Pyflakes A Python source code auditing program which checks for logical errors in the code. . 60, 66

Pylint A Python source code auditing program. . 60, 66

Python A scripting language, designed to be general purpose and easy to read. Home of Python. . 29, 59, 60, 62, 63, 66, 67, 73, 74, 76, 80, 82, 85, 86, 99, 101, 102

raceway A canal made to protect the integrity of cables put into it. . 6

rack A standardized frame or enclosure for mounting equipment in a safe and secure way. . 3–6, 8–11, 14–18, 20, 21, 24, 26–28, 31–34, 37–39, 45–49, 56, 72, 73, 75, 87, 95

RackMonkey A web-based tool for managing racks of equipment. . 28, 29, 51

RackWizard Software that keeps control over inventory in each rack inside SDX . 28, 75

RAM A type of storage that allows individual locations of words or bytes to be directly addressed. The faster technologies for these devices are also volatile, meaning they will lose all stored data when they lose power. . 37, 52

Red Hat A Linux distribution. . 82, 87, 88

refrigerant A substance used in a heat cycle, meaning that the substance is able to pass energy by doing phase transitions. . 23, 24

RHEL A Linux distribution from Red Hat Inc. targed towards the commercial market. . 61

RJ45 A connector commonly used for Ethernet networks. . 34, 37, 84, 89

router A router is a networking device that process and forwards data at the network layer (layer 3) of the Open Systems Interconnection (OSI) model. . 27, 28, 32, 74

RPM A file which contains all the information needed to automatically install the provided software and dependencies providing the dependencies are in a accessible repository. . 61

RRD A type of database which specializes in handle time-series. The database is circular and therefor the file size remains constant over time. . 51–54, 56, 63–65, 68, 82

RRDTool A set of tools designed to handle time-series data and store them in round robin databases. The tool-set was created by Tobi Oetiker. . 80

rST A "what you see is what you get" plain text markup language which can be processed and turned into different usefull formats. . 67

S-403 The room which contains the pumps providing water pressure for the laboratories water circuit. Commonly called the pump room. . 24

SCADA A reference to industrial control systems that monitor and control infrastructure or processes within a manufacturing setting. . 31

Scrum A software development methodology which increase speed and flexibility in projects [20]. . 71, 88, 98

SDX Atlas Trigger and Data Acquisition Room. A building on top of the shaft above the underground experiment. The building which the data center resides within (3178). The building is allocated at Point 1. . 3, 6–9, 11, 13–22, 27, 28, 31, 49, 51–53, 61, 68, 72, 73, 75, 87, 88

securityProbe 5ES One of AKCPs many different devices to monitor an environment. . 36, 38, 40, 49, 51–54, 56, 62, 65, 66, 68, 74

service CDB uses the name service to describe a subnet with the properties of that subnet. Examples are location of devices used in the subnet and alternative **glsdns** names for the devices allowed to get automatic IP addresses from the DHCP servers. . 28

Skype Voice Over IP (VOIP) software used to communicate both verbally and visually. . 97

SLC A distribution made by CERN in the Red Hat family. . 52, 60, 61, 73, 82

SLIMOS The person in charge of security and general safety in SDX. . 16, 17

SMS A way of sending short messages between two units capable of talking over different types of mobile networks. . 7, 16–18, 56, 66

SMTP A protocol that describes how to transfer an email from one computer to another over a network. . 84

SNMP Protocol designed for management of devices on a IP network. . 52, 53, 62, 63, 72, 74, 79, 82, 84, 85, 88, 101

Spectrum A tools to analyze and auto detect hardware on the network. Home of Spectrum. . 101

Spine An SNMP poller written in C and is bundled with Cacti. . 52, 53

Sprint A sprint is a unit in Scrum. Sprints last between one week and one month, and are a time boxed [20]. . 71, 98

Sprint backlog A list of defined tasks that has to be addressed during the next sprint [20]. . 98

SQL A special purpose programming language designed to manage data in relational data base management systems. . 28, 85

SSH A protocol designed for data communication using cryptography to hide the streams content. . 37, 49

subnet A subnet is a part of a IP address space that has been logically divided into a smaller part. . 27, 28, 88

Subversion An open source revision control software package which is used to control the development of software. . 60, 61, 76

switch A switch is a networking device that process and routes data at the data link layer (layer 2) of the OSI model. . 28, 32, 74

Sys-admin The system administrators is the team responsible for installing and maintaining the machine park in Atlas. . 17, 27, 49, 71

SystemV A initialization implementation that takes care of automatically start and shut down services (daemons) at boot, restart or shutdown. . 61

TDAQ Collects data from detectors and filters out the events of interest using software trigger algorithms. . 8

Telnet A network protocol used on any type of network to provide bi-directional interactive text-oriented communication. . 37

thermistor A type of resistor which has a varying resistance depending on device temperature. Widely used as inrush current limiters because of its property to lose resistance as it is getting warmer. . 11

thread A thread is a light-weight process managed by the process that instantiated the thread. Threads instantiated by the same process share the same resources within the program. . 62–66, 74

TMP00 A temperature sensor designed for measuring the temperature of airflow. Connects to one of the 8 RJ45 ports on a node. . 40, 42, 43, 45, 47, 81, 89

TMP01 A temperature sensor with the same specifications as TMP00 and DCT00, but it does not have a casing making it more sensitive. . 37, 40–43, 45, 48

U A rack unit or U is a unit of measure used to describe the height of equipment intended for mounting in a rack. 1U is 1.75 inches. . 5, 6,

8–11, 20, 47

UPS An uninterruptible power supply that provides emergency power to a load when the input power source, typically mains power, fails. . 3, 4, 13–15, 33, 34, 38, 49

V A derived unit of electric potential (Volt) in the international system of units (SI). . 9, 10, 20

Velcro A brand name of a commercially marketed fabric hook and loop fastener. . 6, 45

Vim An open source text editor based on vi and written by Bram Moolenaar. . 60

Virtualbox An open source hypervisor designed to run guest OSs. . 52, 60, 66

VLAN A concept of separating a physical network into two or more virtual network. . 27, 28

VME A videly used bus for industrial purposes. . 5

VOIP A family of technologies, methodologies and communication protocols. . 88

W A derived unit of power in the international system of units (SI). . 8–11, 13, 14, 20, 22, 24

Windows An OS created by Microsoft in the 80s. . 21

XP A software development methodology which is intended to improve software quality and responsiveness to changing customer requirements[20]. . 98

XPU Has the network connectivity to either function as a level 2 filter processor or as a level 3 filter processor. . 17

Bibliography

[1] AKCP. daisyTemp Sensor - DCT00. http://..., 2012.

[2] AKCP. Temperature Sensor - tmpxx. http://..., 2012. [Online; accessed 28-August-2012].

[3] Allbery, B. S., Bostic, K., Eckhardt, D., Faith, R., and et al. Filesystem hierarchy standard. http://..., 2004. [Online; accessed 05-December-2012].

[4] Armstrong, S., Barczyk, M., Bogaerts, J., Boisvert, V., Burckhart.Chromek, D., Ciobotaru, M., and et. al. Atlas high-level trigger, data acquisition and controls - technical design report. Tech. rep., CERN, 2002. [Online; accessed 08-July-2012].

[5] Beck, H., Dobson, M., Ermoline, Y., Francis, D., Joos, M., Martin, B., Unel, G., and Wickens, F. Atlas DAQ/HLT Infrastructure. Tech. rep., CERN, 2005. [Online; accessed 08-July-2012].

[6] Carrier. 30RB AQUASNAP. http://... [Online; accessed 5-August-2012].

[7] Dobson, M., Francis, D., Gorini, B., Martin, B., and et. al. Atlas TDAQ Cooling Failure. [Received on E-mail by Brian Martin], 2009.

[8] Hoff, K. G. *Bedriftens Økonomi*, 7 ed. Aschehoug, 1998.

[9] Jesse Noller. Python threads and the global interpreter lock. http://..., 2009. [Online; accessed 18-November-2012].

[10] Jesse Noller, and Richard Oudkerk. Addition of the multiprocessing package to the standard library. http://..., 2009. [Online; accessed 18-November-2012].

[11] M, K. Application note, dew-point calculation. Tech. rep., Sensiron AG, 2006. [Online; accessed 03-December-2012].

[12] Martin, B. NIM Infrastructure. Tech. rep., CERN.

[13] Martin, B. Codes of Practice in SDX. Tech. rep., CERN, August 2005.

[14] Martin, B. Lab 4 infrastructure. [Reveiced on E-mail by Brian Martin], 2012.

[15] NTI. Enterprise Server Environment Monitoring System. http://..., 2012. [Online; accessed 13-October-2012].

[16] Python Software Foundation. Threading – higher level threading interface. http://..., 2012. [Online; accessed 18-November-2012].

[17] Savu, D., Martin, B., Sjøen, R., Batraneanu, S., Stancu, S., and Al-Shabibi, A. Efficient network monitoring for large data acquisition systems. Tech. rep., CERN, 2011. [Online; accessed 27.10.2012].

[18] Shlomo Novotny. Green Field Data Center Design - Water Cooling for Maximum Efficiency. Tech. rep., Vette Corp, 2010. [Online; accessed 08-July-2012].

[19] Snevely, R. *Enterprise data center design and methodology*, first ed. Prentice Hall Press, Upper Saddle River, NJ, USA, 2002.

[20] Sommerville, I. *Software Engineering*, 9 ed. Pearson, 2010.

[21] Wikipedia. Emergency power system — wikipedia, the free encyclopedia. http://..., 2012. [Online; accessed 12-December-2012].

[22] Wikipedia. Rack — wikipedia, the free encyclopedia. http://..., 2012. [Online; accessed 12-December-2012].

[23] Wikipedia. Thermistor. http://..., 2012. [Online; accessed 12-December-2012].

[24] Wikipedia. Uninterruptible power supply. http://..., 2012. [Online; accessed 12-December-2012].

[25] Wiremold, L. A New Standard for Wire and Cable Management Systems. http://..., 2012. [Online; accessed 08-March-2012].

[26] Youngm Hugh, Freedman, R., and Ford, L. *University Physics with Modern Physics*, 12 ed. Pearson Addison-Wesley, 2007.

Appendices

Preproject - Goals and Boundaries

.1 Project Goals

.1.1 Task definition / Intended results

My task in this project will be to take a leading role in a project which main goal is to design, implement and document a test-bed system in building 4 at CERN.

The intentions are to:

- commission the test-bench.
- port existing tools from the Atlas environment, reconfigure and commission them for stand alone use in the lab.
- co-ordinate with the sys-admin group to define areas of responsibility.
- introduce environmental monitoring and remote power on/off functionality at the rack level.
- fully document the test-bench and automate regular consistency checks to confirm that the documentation reflects reality.

.1.2 Intended effect

To provide a fully equipped documented and supported test-bench to

- offer software development platforms.
- provide some basic services.

- give researchers access to reconfigurable processor and routing facilities.

.1.3 Personal goals

I have defined a set of personal goals for me to achieve while here at CERN. They may be or not be relevant to this thesis, but I wish to specify them here to give myself some specifics to reach.

- Get an understanding of how data centers are designed and built.
- Get an understanding of how to do maintenance service on and in the data center.
- Get a better understanding of networking in general.
- Raise my standard of writing technical papers.

.2 Conditions and Constraints

There will be conditions and constraints from both Gjøvik University College and CERN. Both parties conditions will be respected and followed. In cases where this is not possible both parties will be contacted to find a solution.

.2.1 Gjøvik University College

The rules to follow will be:

- Rules and requirements defined by Gjøvik University College.
- Rules indicated by the Bachelor thesis at CERN[2] homepage.
- Time limits defined by my supervisor in Norway.

.2.2 CERN / Atlas

The rules to follow will be:

- Rules and requirements defined by CERN, Atlas and my supervisor.

[2] http://english.hig.no..TØL3903

- Time limits defined by my supervisor.
- Rules and guidelines put forward by the Safety Information Registration[3] courses at CERN.
  - Basic Safety (Levels 1 and 2)
  - Specific Risks (Level 3)
  - Computer Security
  - LHC Machine (Level 4)
  - Atlas Safety (Level 4A)
  - Electrical Safety Awareness
- Codes of Practice in SDX [13].

.3 Work process

A blog has been created[4] that will while this thesis is being written be updated with project updates and relevant information. Although the blog is not a requirement, it is a requirement for most other bachelor thesis at the university and I would like to adhere to this convention.

.3.1 Follow-up and reporting

An agreement have been made to talk with my supervisor Erik Hjelmås over Skype every second Friday at 10:30 or when needed. My main focus here will be to make sure that all parties is getting the information they want and also to make sure there is someone to help me when help is needed.

Since my direct supervisor here is also my team leader, daily contact will be kept.

[3] http://sir.cern.ch
[4] http://cern.moshwire.com

.3.2 Work methods

I have been integrated into an existing group and project and I will have to follow the groups existing methods and work style. However, project management tools can still be used to give me some guidance for this thesis.

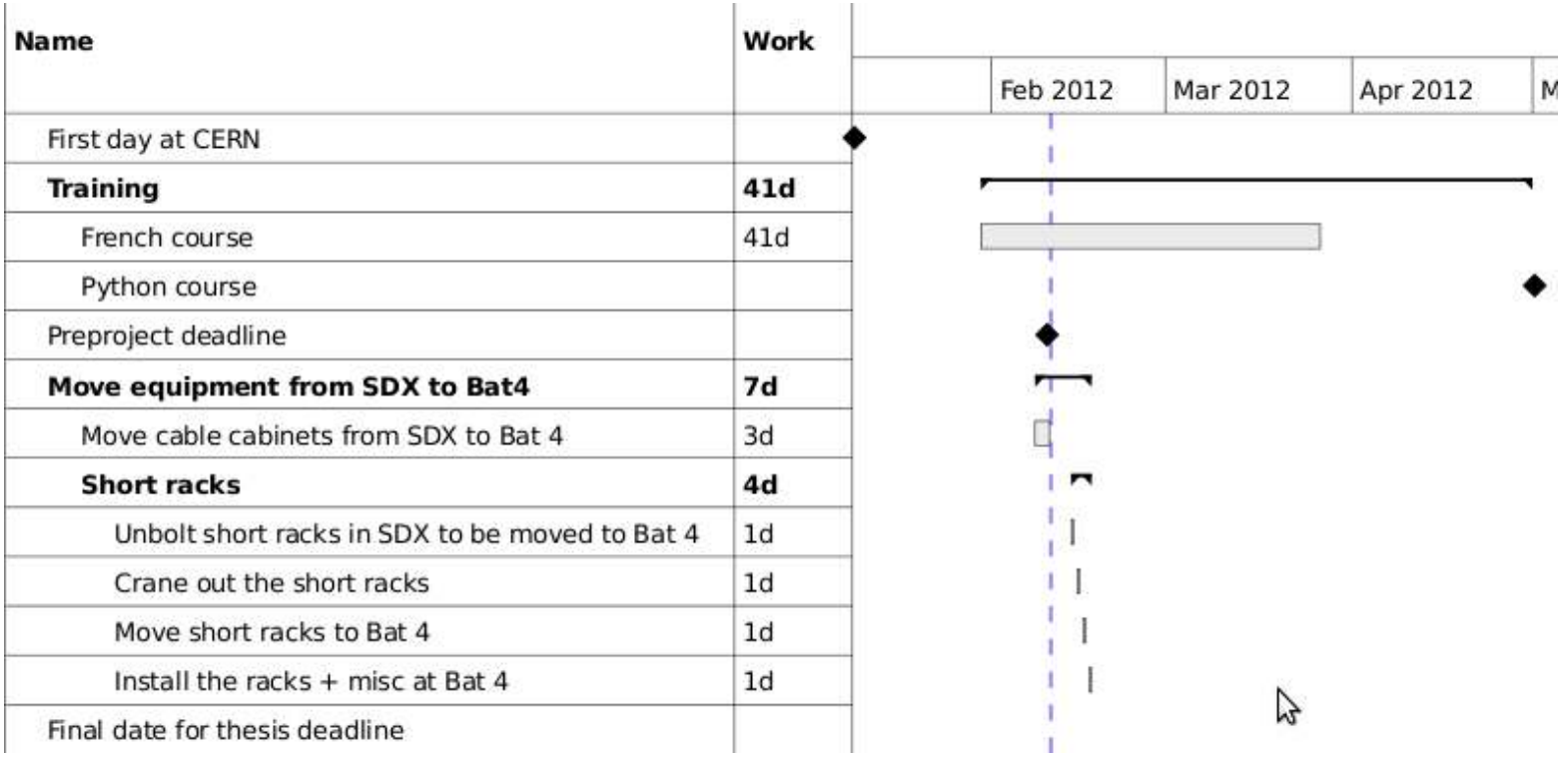
Implementing a project management tool like Scrum, Extreme Programming (XP) or similar in full scale will make things more complex than needed when there is only me working on the thesis. However, using bits and pieces from a management tool will give me some structure and protect me from straying to much from the intended path.

Four points have been chosen from Scrum to give me a little guide ability and structure. The reason Scrum has been chosen is that I have more experience with it and it has some rules, that are simple enough to not over complicate the process [20].

I will -

- - use Sprints of two weeks.
- - create a product backlog together with my supervisor.
- - try to use Sprint backlogs and see if this provides me with increased guide ability. The reason for choosing to just try is that I am not sure about how much work per 2 weeks there will be.
- - meet up with my supervisor for a review and planning meeting every second Friday.

.3.3 Gant Scheme



Some notes about the schedule

- When the French course is finished, Brian will apply for me to attend the next Python course at CERN. So there is no specific date set yet.

.3.4 Risk analysis

All through this project there will be several risks that needs to be addressed. For some of the risks it will be enough to just be aware that the problem can occur and for others a contingency plan must be created.

Problem / Risk	**Severity**	**Odds**	**Contingency plan**
might get sick	Low	Medium	Consider how severe the sickness is and try to plan ahead
The thesis can not be finished	High	Low	Try to find a solution with my supervisor with the work all ready created
Work on the Atlas project takes all my time	Medium	Medium	Talk with my supervisors to find a solution where I am able to do the required work or change the project definition so that the thesis can be finished through the work done on the Atlas system.
Test-bed project get shut down	High	Low	Contact my supervisors and find a solution.
The requirements for the server room is changed in a drastic way	Medium	Low	Together with my supervisors, change the project requirements or change the task definition
The project can not be finished by the thesis deadline	Low	High	When doing the review and planning meeting. An estimate should be made to see if the work can be finished in time for my thesis deadline, if not, start working on preparing the work done for the person taking over my responsibilities in the project.

.4 Tools and training

.4.1 Tools

Tools that will be actively used in this thesis.

- LaTeX
  The thesis will be written in this document preparation system.
- Bibtex
  Tool to process references in a TeX document.
- Python
  Scripting tools for the system.
- Git revision control
  Keeping control over the project files and extra backup on a central machine.
- Planner
  Project Management Application for Linux.
- lanDB / CDB
  Databases containing information about the network and hardware on the different systems at CERN.
- Spectrum
  A tool to analyse and auto detect hardware on the network.
- Net-LOG
  A tool created in Python by CERN to report SNMP and Spectrum errors on the network in a processed and controlled way.
- Net-IS
  An investigation tool created in Python by CERN to investigate and monitor the general behaviour of the Atlas network, both through specific data collection and through graphical representations.
- Open Office Draw
  A tool to create sketches, diagrams and more.

.4.2 Training

It has been made very clear that I will have to do a lot of self study and that I also will be taking courses to better my ability to do my job.

It has been decided for now that I will attend two courses available through CERN.

- Beginners course in French.
  I have no prior knowledge of French and it has been clear that I would benefit from a basic understanding. I have all ready begun this course, three times a week for two hours in each session until the 5th. of April. Attendance in class is obligatory and at the end of the course, I will have to pass an exam.
- Python.
  CERN provides a Python course for all its employees that need it. I will be attending it when the French course is done.

Topics that will be studied on my own are:

- Networking.
- Environmental measurements and e-reporting.
- Scripting for power control in the test-bench.

Ciat - 47U/52U-5v

Client :

Référence :

Date : 19/09/08

Page : 1

Refroidisseur de rack - CLIMRACK série CR

1 - DESCRIPTIF TECHNIQUE

Climatiseur de rack CR 52-5v comprenant :

20 x Porte arrière de marque RITTAL
- RAL 7035
- 600 de large, hauteur 52U
- *Porte arrière rack RITTAL TS8 fournie par le CERN*

Equipée de son refroidisseur comprenant
- 1 Batterie tube cuivre ailette aluminium à faible perte de charge aéraulique
- 1 capot de protection de la batterie à la couleur de la porte (RAL 7035)
- 1 grille de protection de l'échangeur RAL 7035
- 5 ventilateurs axiaux basse consommation équipés d'une sortie tachymétrique
- (MTBF = 40 000 H)
- 1 carte de gestion alimentation, défaut et raccordement client large plage alimentation 110V à 230V
- Câbles sans halogène entre moteur et carte de gestion

2 - CARACTERISTIQUES TECHNIQUES

Performances

T° / Hygrométrie à la reprise	Débit d'air (m3/h)	Régime d'eau (°C)	Puissance (kW)	Débit d'eau (m3/h)	T° / Hygrométrie au soufflage	Perte de charge batterie (mCE)
36°C / 10 g/Kg air sec	3 800	14 / 20	14.4	2.07	24.6°C / 10 g/Kg air sec	0.968

- **Limite de fourniture :**

  La présente proposition ne comprend pas :
  - La fourniture des PC
  - La fourniture des RAILS de montage
  - La fourniture des pièces spécifiques pour le montage des PC avec leurs racks (exemple équerre…)

Matériel identique à ARC SO 9022511 du 30/10/2009

TELECOPIE

EMETTEUR

CIAT Agence Savoies - Ain
Le Mont Fleuri 9. av du Pré de Challes
P.A.E Les Glaisins 74940 Annecy - le - Vieux
Téléphone 04.50.64.02.75
Télécopie 04.50.64.03.16

DESTINATAIRE

SOCIETE: .....CERN.....01 Prevessin .....

Télécopie: .....00.41.22.0000000..... Téléphone :.....00.41.22.0000000.....

Mr .....Youri ERMOLINE ..... Réf.:.....Dossier ATLAS / TDAQ.....

Date : .....12.../..05.../..2004..... Nombre total de page : .....

De la part de : Thierry CANAL

PROPOSITION N° : 125 / 04 / 0685 / TC

Monsieur,

Suite à notre dernière rencontre et aux éléments que vous nous avez fournis , veuillez trouver ci-joint notre offre pour la fourniture des différents produits :

Baies racks + Aéroréfrigérant

- Ensemble batterie de refroidissement CIAT + Groupes moto-ventilateur type **SPN 82148 EC**

1 x Batterie SPN 82148EC
2 rangs tubes cuivre – 8 circuits – 2 passes
ailettes aluminium pas 1,6 mm
1 x Ensemble de 3 groupes moto-ventilateur montés sur tôlerie batterie

Puissance frigorifique = 9,5 Kw

Suivant même définition thermique jointe.

- Ensemble armoire pour mise en place de PC industriel
- Flux d'air horizontal, entré par face avant (sans porte) et extraction par unités de refroidissement aéroréfrigérante SPN montés sur la face arrière
- Limite de fourniture :
  La présente proposition ne comprend pas :
  - La fourniture des PC
  - La fourniture des RAILS de montage
  - La fourniture des pièces spécifiques pour le montage des PC avec leurs racks (exemple équerre…)

Désignation	**Quantité**	**P.U. H.T.**	**Montant total H.T (euros)**	**Montant total H.T (Fr Suisse)**
VERSION PROTOTYPE **1 x Rack Baie PC 19 ″ - RITTAL** **Hauteur 47U** (Ht 2200+100) x largeur 600 x profondeur 1000 mm	1	3 392,00 €	3 392,00 €	
1 x Batterie SPN 82148EC **1 x Ensemble tôlerie prélaquée avec 3 groupes moto-ventilateurs montés sur la batterie** (raccordement électrique sur bornier accessible côté intérieur porte) Refroidisseurs à débordement extérieur de 150 mm environ et débordement intérieur minime	1 1	Inclus Inclus	Inclus Inclus	
1 x Extension de garantie pièces à 2 ans des groupes moto-ventilateurs	1	Inclus	Inclus	
1 x Mise en place aéro (batterie + groupes moto-ventilateurs) sur porte arrière Rack	1	Inclus	Inclus	
1 x Bâtit	1	Inclus	Inclus	
1 x Enjoliveur face avant 1 jeux	1	Inclus	Inclus	
2 x Montant 19 ″ (avant et arrière) renforcé	1	Inclus	Inclus	
1 x Porte arrière	1	Inclus	Inclus	
2 x Panneaux latéraux	1	Inclus	Inclus	
1 x Toit	1	Inclus	Inclus	
1 x Socle H100	1	Inclus	Inclus	
1 x Tôles de fond	1	Inclus	Inclus	
1 x Kit visserie	1	Inclus	Inclus	
1 x Montage des éléments ci-dessus	1	Inclus	Inclus	
1 x Câblage des ventilateurs avec leur protection	1	Inclus	Inclus	
OPTIONS	**Quantité**	**P.U. H.T.**	**Montant total H.T**	
3 x Détecteur de rotation avec signal brut ramené sur bornier (1 par ventilateur)	1	180,00 €	180,00 €	
Face Avant 1U – **RITTAL**	1	6,52 €	6,52 €	
Face Avant 2U- **RITTAL**	1	9,95 €	9,95 €	
Face Avant 3U - **RITTAL**	1	12,20 €	12,20 €	
Face Avant 4U - **RITTAL**	1	16,59 €	16,59 €	
TRANSPORT ET EMBALLAGE	**Quantité**	**P.U. H.T.**	**Montant total H.T**	
Frais de transport (Culoz / Prevessin)	1	270,00 €	270,00 €	

NOTA : SANS REGULATION

N° : RB 03 09 35

Date émission : 09/07/2012
Page : 1 / 5

Client : CERN
Contact :
Votre référence : Refroidisseurs racks 19"

Agence : Belley
Votre interlocuteur :
Téléphone : 04 79 81 68 23 Fax : 04 79 81 15 49
e-mail : R.Bruyère@ciat.fr

BATTERIE D'ÉCHANGE SPN 82160 EC

Ailettes gaufrées à haute performance et peu sensibles à l'encrassement
Caractéristiques et encombrements suivant notice technique N...

PUISSANCE THERMIQUE	9,5 kW	
Surface d'échange	26,5 m²	
	INTERIEUR TUBES	EXTERIEUR TUBES
FLUIDE	Eau	Air
Pression d'entrée	0,9 MPa abs.	101 300 Pa abs.
Débit	2 000 kg/h	2 450 m³/h
Référence débit		20 °C / 50 %(HR)
Vitesse moyenne	1 m/s	1,71 m/s
Entrée	15 °C	33 °C / 10 g/kg Air sec
Sortie	19,1 °C	21,5 °C / 10 g/kg Air sec
Perte de charge	11,7 kPa	2,16 DaPa

CONSTRUCTION / DIMENSIONS

Position D090: Veine d'air Horizontale, Tubulures à Droite (*)	Haut. Ailetée : 256 mm
Circulation par coudes et collecteurs	Long. Ailetée : 1600 mm
Tubes: Cuivre 9,52 x 0,35	Nb de rangs : 2
Ailettes: BG0932Q1 Aluminium	Pas : 1,6 mm
- Entrée: Raccord Fileté 1"	Nb de circuits : 8 (2P)
- Sortie: Raccord Fileté 1"	Volume : 2,25 dm³

DESP 97/23/CE: Article 3.3 - Press. / Temp. maxi admissible: 15 bar eff. / 110 °C
Tôlerie Acier galvanisé — Poids à vide du faisceau : 28,6 kg

- Répartition du débit d'air sur 3 groupe-motoventilateurs au standard européen avec grille de protection moteur
- Alimentation électrique mono 230V-50hz , accès côté intérieur rack, avec connexion sur bornier unique aux 3 groupe-motoventilateurs, liaison par câble non halogène
- Alimentation d'eau par l'intérieur du rack filetage pas du gaz ou métrique
- Ouverture de la porte à 110°

Dimensions approximatives du refroidisseur
Débordement extérieur : Epaisseur 150 mm – Hauteur environ 1800 mm - Largeur environ 380 mm
Débordement intérieur : environ 60 mm
Poids total faisceau + tôlerie + ventilateurs : kg environ

Nota : connexions hydrauliques:

Idem affaire précédente de Mr RACZ

Nota : garanties:
Des ventilateurs :
Garantie des appareils 2 ans pièces à compter de la mise à disposition (voir plus-value sur groupes moto-ventilateurs)

Faisceau ailetté des batteries SPN
Garantie des appareils 2 ans pièces à compter de la mise à disposition sur les batteries SPN
Cependant, nous vous rappelons qu'en cas d'utilisation de fluides non compatibles avec les matériaux utilisés (tubes cuivre, ailettes aluminium) notre responsabilité ne saurait être engagée en cas de corrosion.

Nous restons à votre disposition pour tout renseignement complémentaire.
Sincères salutations.

Délai :
Fourniture des RACKS + REFROIDISSEURS : 8 semaines hors congés (plus temps de transport Culoz / Prévessin)

Nota un délai plus précis sera communiqué après engagement du fournisseur.

DEPARTEMENT INDUSTRIE
T. CANAL

CONDITIONS DE PRIX :
Nos conditions s'entendent pour une commande globale :
- **UNITAIRES - HORS TAXE - REMISE DEDUITE.**
- **FRANCO DE PORT ET D'EMBALLAGE, matériel non déchargé – Livraison France – 01 Prevessin**
- En cas de fractionnement, les conditions seraient à revoir.
- Base tarif : TARIF CIAT HT de JANVIER 2004 - Validité des prix : 1 mois.

AKCP - Monitoring equipment

AKCess Pro
http://www.akcp.com

AKCess Pro Limited
38th Floor Central Plaza
18 Harbour Road, Wanchai
Hong Kong

QUOTATION
No.: 102200

Quotation to:
CERN
Attn: Hans Åge Martinsen
CH-1211 Geneva 23
Switzerland

Quotation Date: 06/25/2012

Customer e-mail: hmartins@cern.ch

Description	Product Code	Qty	UoM	List Price	Disc %	Price	Line Amt
securityProbe 5E standard - NO video capability (AKCP)	SEC5ES	1	Ea	995.00		995.00	995.00
E-sensor8 Expansion w/ 5ft cable, SEC only (AKCP)	E-IS8	2	Ea	300.00		300.00	600.00
IO-digital8 via 8 x 2 pin connectors include 5ft cable	IODC8	2	Ea	65.00		65.00	130.00
daisyTemp 8 pack daisy-chainable temperature sensors (AKCP)	DCT00-8	1	Ea	500.00		500.00	500.00
TCAJ Thermocouple Adapter (AKCP) (or TCAK)	TCAJ	8	Ea	125.00		125.00	1,000.00
Shipping and Handling Charge - box E - weight about 7 kg		1	Ea			155.00	155.00
Standard						3,380.00	0.00

Grand Total (USD): 3,380.00

Amount in Words: USD ThreeThousand-ThreeHundred-Eighty 00/100

Terms & Conditions
100% Advance Payment, delivery after payment is received.
For TT payment, please note that you are responsible for payment of your local bank fees and the oversea cable charges.
This quotation is valid for: 30 days

Shipping Details
CERN
Hans Åge Martinsen
CH-1211 Geneva 23
Switzerland

Notes: Our price does not include any import customs duties, VATs and other local government taxes.

Wide Technology Partners AG Bahnhofplatz 6300 Zug

CERN
Brian Martin, Dep PH
1211 Genève 23

Wide Technology Partners AG
Bahnhofplatz, 6300 Zug
T +41 (0) 41/240 49 49
F +41 (0) 41/240 49 59
info@wide.ch / www.wide.ch

Offer No. 01-001755
Version:1.1

Zug, 27.06.2012

Article Code	Description	Qty.	Unit	Price/Unit	Amount CHF
	AKCP				
SEC5ES	securityProbe 5E standard (no video capability)	1		965.00	965.00
E-IS8	E-Sensor8 Expansion unit with 5Ft cable, use with SEC5ES	2		295.00	590.00
IODC8	IO-digital8 via 8 x 2 pin connectors include 5ft cable	2		63.50	127.00
DCT00-8	AKCP daisyTemp 8 pack daisy-chainable temperature sensors	1		487.00	487.00
TMP00	Temperature sensor with free 5 feet cable	8	pcs.	73.00	584.00
Transport/Zoll	Shipping&Handling Charge (plus customs charges upon receipt if applicable)			155.00	155.00
	TOTAL Subject VAT-free				2'908.00

Contact: Brian Martin
Validity: 20 days
See our general terms and conditions for further reference (www.wide.ch).

Order Confirmation:
Name: _____
Date: _____

Signature: _____

Sketches of suggested environmental monitoring solutions

Figure 1: *First suggestion of environmental monitoring with NTI*

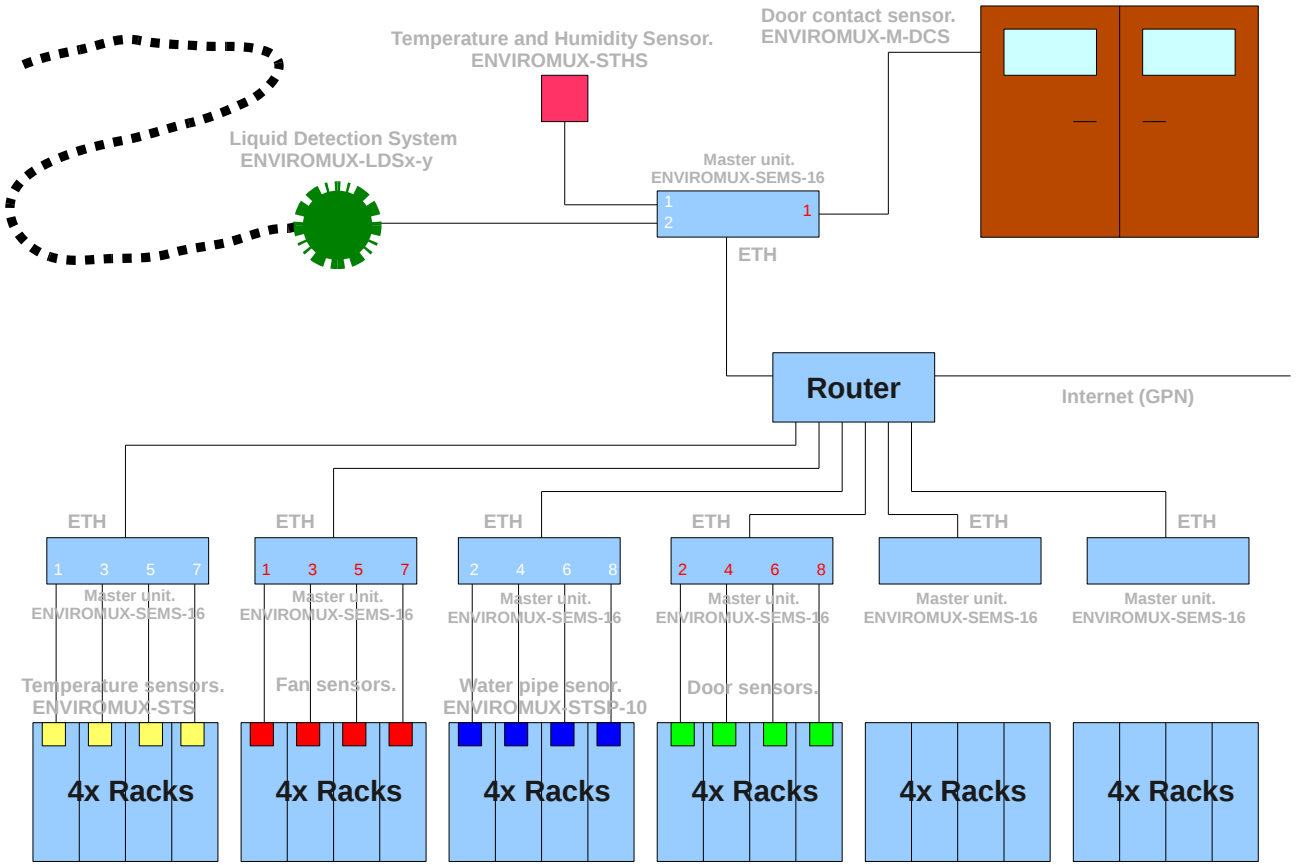


Figure 2: *Second suggestion of environmental monitoring with Jacarta*

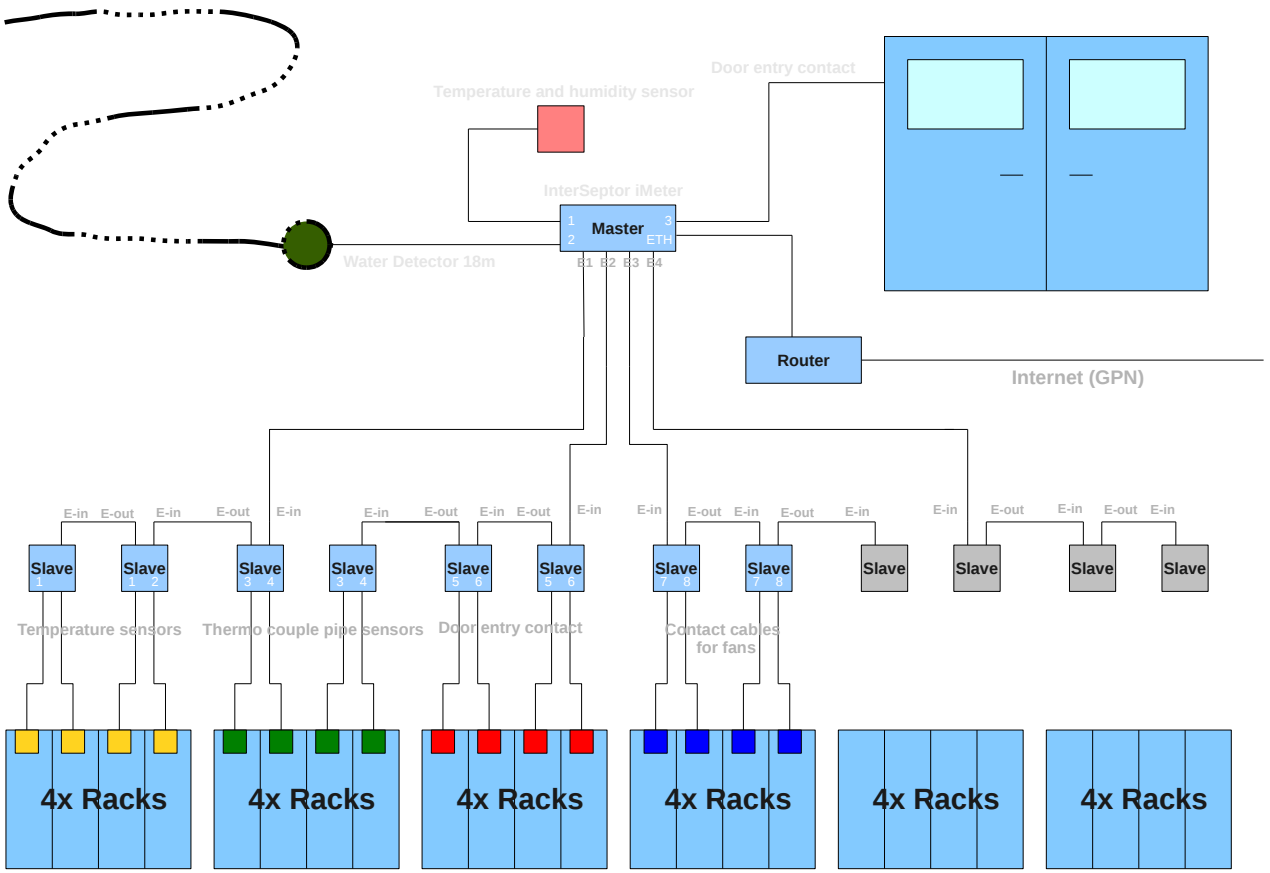
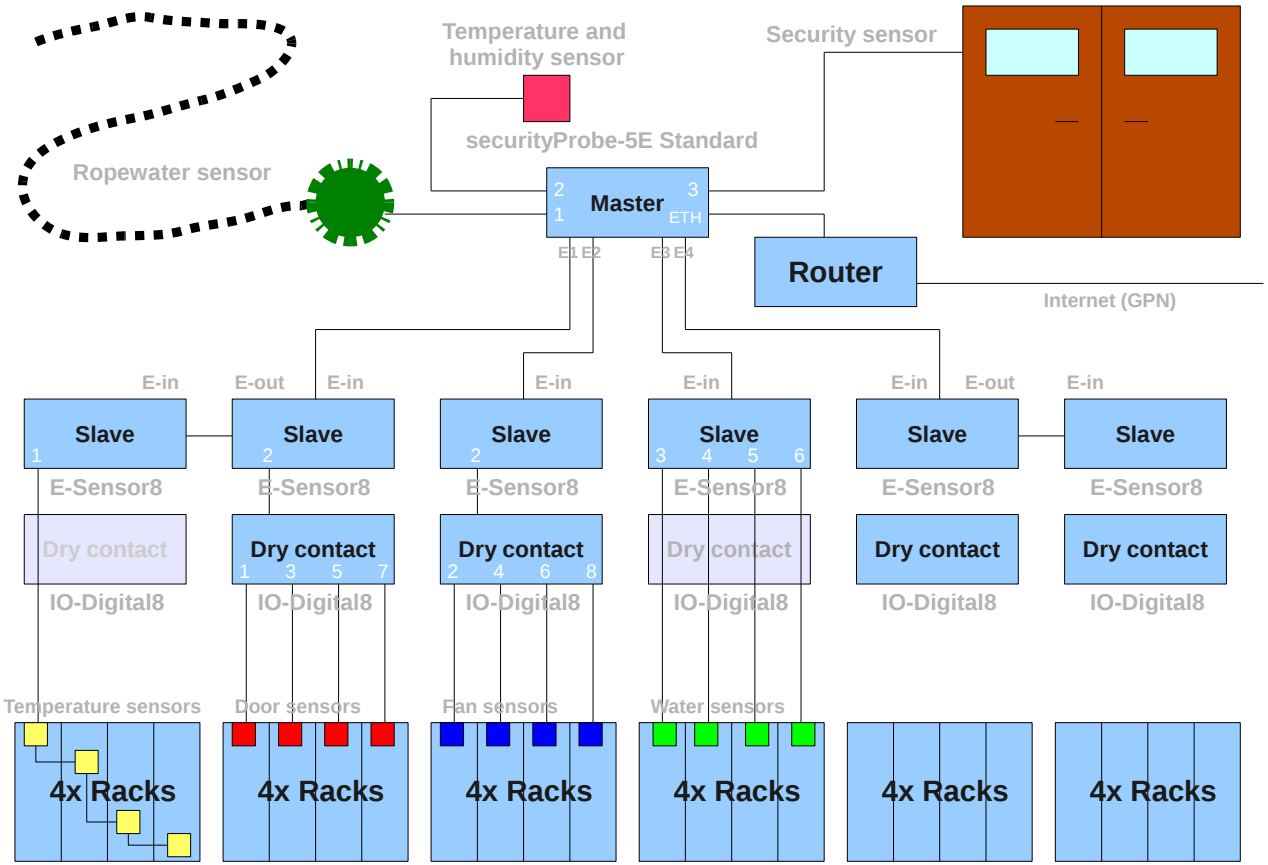


Figure 3: *Third suggestion of environmental monitoring with AKCP*



Calibration

Sheet1

TMP01 Temperature Sensor									
Sensor	**0C**	**5C**	**10C**	**15C**	**20C**	**25.5C**	**30C**	**35C**	**40C**
TMP01/05/16/12-05	00.00	04.50	09.50	15.00	20.00	25.50	30.00	35.00	40.00
TMP01/05/16/12-03	00.50	05.00	10.50	15.50	20.00	25.50	29.50	35.00	39.50
TMP01/08/09/12-03	00.00	04.50	09.50	15.00	19.50	25.50	29.50	34.50	39.50
TMP01/08/09/12-04	00.00	04.50	09.50	15.00	19.50	25.50	30.00	35.00	40.00
TMP01/08/09/12-02	00.00	04.50	10.00	15.00	20.00	25.50	30.00	35.00	40.00
TMP01/05/16/12-06	00.50	04.50	09.50	15.00	20.00	25.50	30.00	35.00	40.00
TMP01/08/09/12-01	00.00	04.50	10.00	15.50	20.00	25.50	30.00	35.00	40.00
TMP01/05/16/12-04	00.00	05.00	10.00	15.50	20.00	25.50	30.00	35.00	40.00
Average	00.13	04.63	09.81	15.19	19.88	25.50	29.88	34.94	39.88
Sensor	**45C**	**50C**	**55C**	**60C**	**65C**	**70C**	**75C**		
TMP01/05/16/12-05	45.00	50.00	55.00	60.50					
TMP01/05/16/12-03	45.00	50.00	55.00	60.00					
TMP01/08/09/12-03	44.50	50.00	55.00	60.00					
TMP01/08/09/12-04	45.00	50.00	55.00	60.50					
TMP01/08/09/12-02	45.00	50.50	55.50	60.50					
TMP01/05/16/12-06	45.00	50.50	55.50	60.50					
TMP01/08/09/12-01	45.00	50.50	55.50	60.50					
TMP01/05/16/12-04	45.00	50.00	55.50	60.50					
Average	44.94	50.19	55.25	60.38	00.00	00.00	00.00		

Em-poller Error Messages

Figure 4: *Example email received when a sensor can not be polled.*

```
Date: Tue, 27 Nov 2012 14:28:38 +0100
From: netadmin@cern.ch
To: atlas-tdaq-networking@cern.ch
Subject: Sensor TD09 WATEREGRESS has gone into an error state.

Hello network administrators!

    This is an automated notification that the sensor TD09 WATEREGRESS
    in the laboratory has gone into an error state. The reason is
    unknown.

    Cheers,
    netadmin@cern.ch
```

Figure 5: *Example email received when a sensor is reporting temperatures above critical levels.*

```
Date: Wed, 28 Nov 2012 15:20:59 +0100
From: netadmin@cern.ch
To: atlas-tdaq-networking@cern.ch
Subject: Sensor TD09 AIR has changed state.

Hello network administrators!

    This is an automated notification that the sensor TD09 AIR
    in the laboratory has changed error state.

    Sensor location: TD09 AIR
        Error message: 'Critical temperature: 32.5C'

    Cheers,
    netadmin@cern.ch
```