

Azure Automation State Configuration

Øyvind Lotten Westrum

Agenda

- Beskrivelse av oppgaven
 - Intro til Azure Automation State Configuration
 - Demo
 - Konklusjon
 - Spørsmål
-
- Det er bare å stille spørsmål underveis

Oppgaven

Utvikle et sluttprodukt (Proof of Concept) i form av en ferdig konfigurert løsning for konfigurasjon av “Session Hosts”

Løsningen skal automatisk konfigurere “Session Hosts” uten manuelle steg og sikre at konfigurasjonen utilsiktet ikke blir endret over tid.

Oppgaven skal løses med Azure Automation State Configuration (Azure DSC)

Effektmål for prosjektet

- Øke grad av automatisering ved å benytte Azure DSC for installasjon, konfigurasjon og drift av servertjenester på Windows-plattformen.
- Bedre kvaliteten på tjenestene ved å unngå avvik og konfigurasjonsdrift.

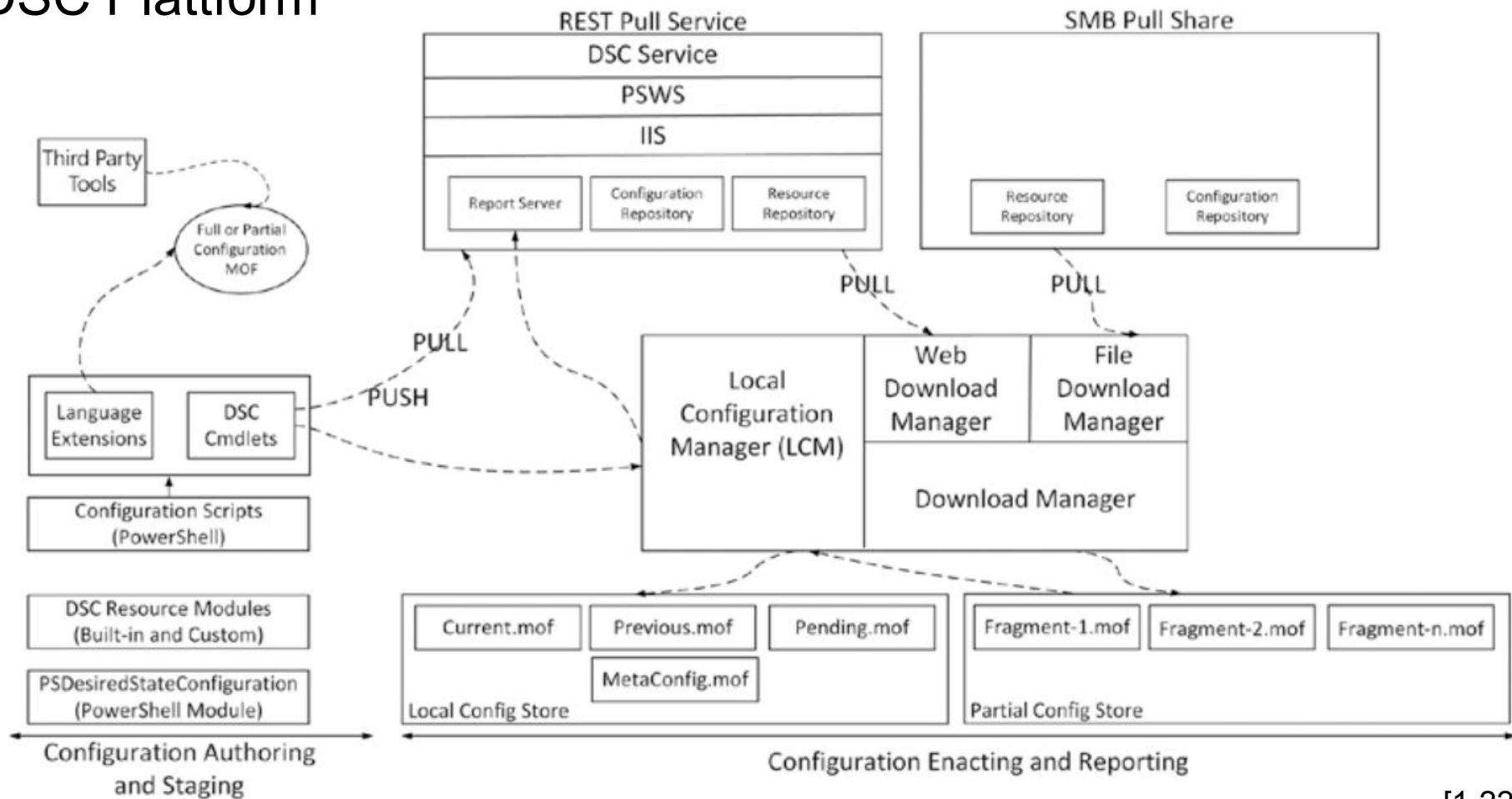
Motivasjon

Microsoft har anbefalt brukere å starte overgangen til Azure DSC siden den lokale Pull serveren ikke lenger vil videreutvikles med ny funksjonalitet

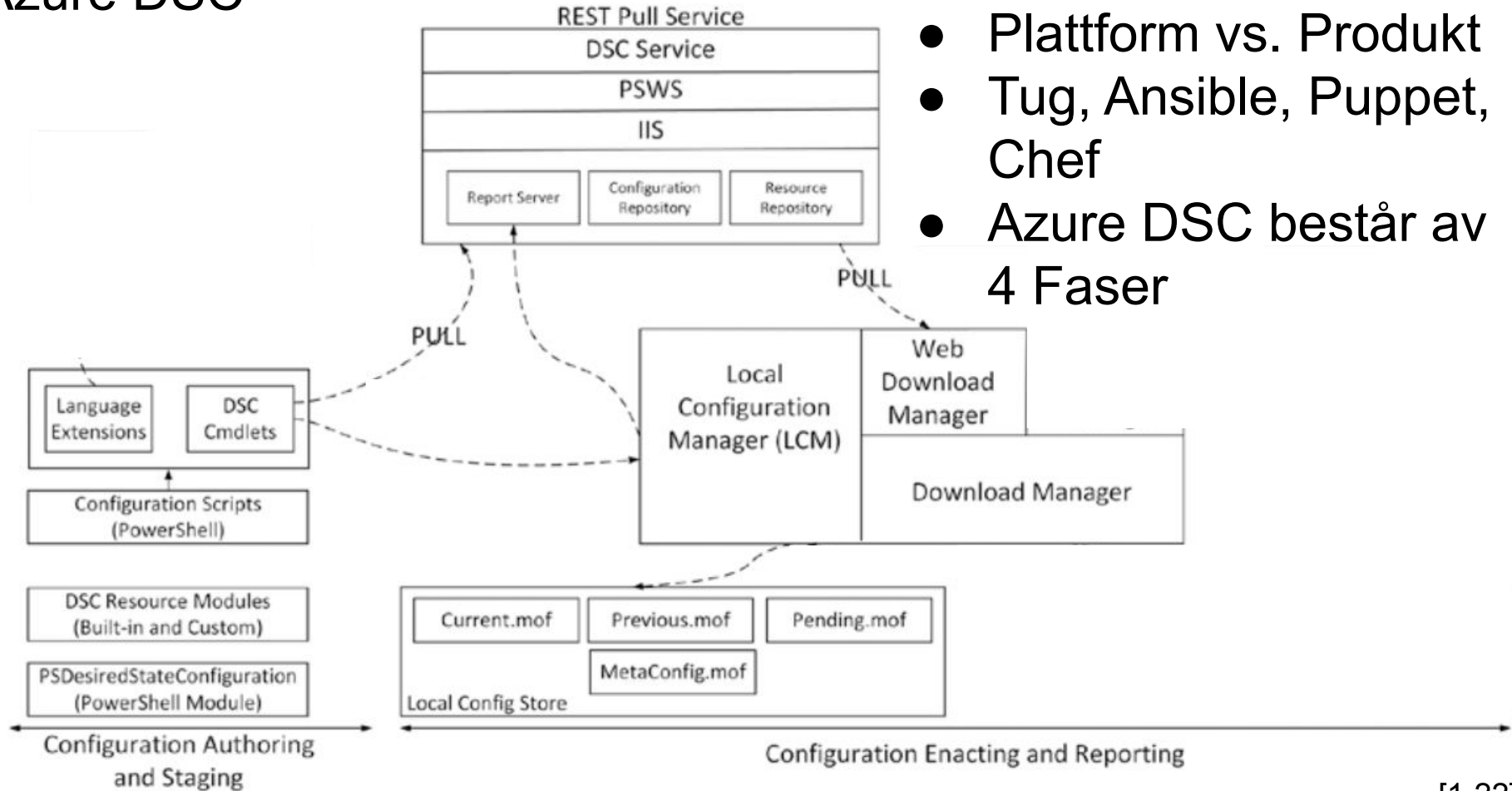
Azure Automation State Configuration

- Mulighet for å konfigurere Windows-Servere automatisk
- Konfigurasjonsdrift
- Programmerbar infrastruktur
 - Sikrer og forenkler endringsprosessen
 - Selvdokumenterende
 - Lar deg fokusere på andre oppgaver

DSC Platform

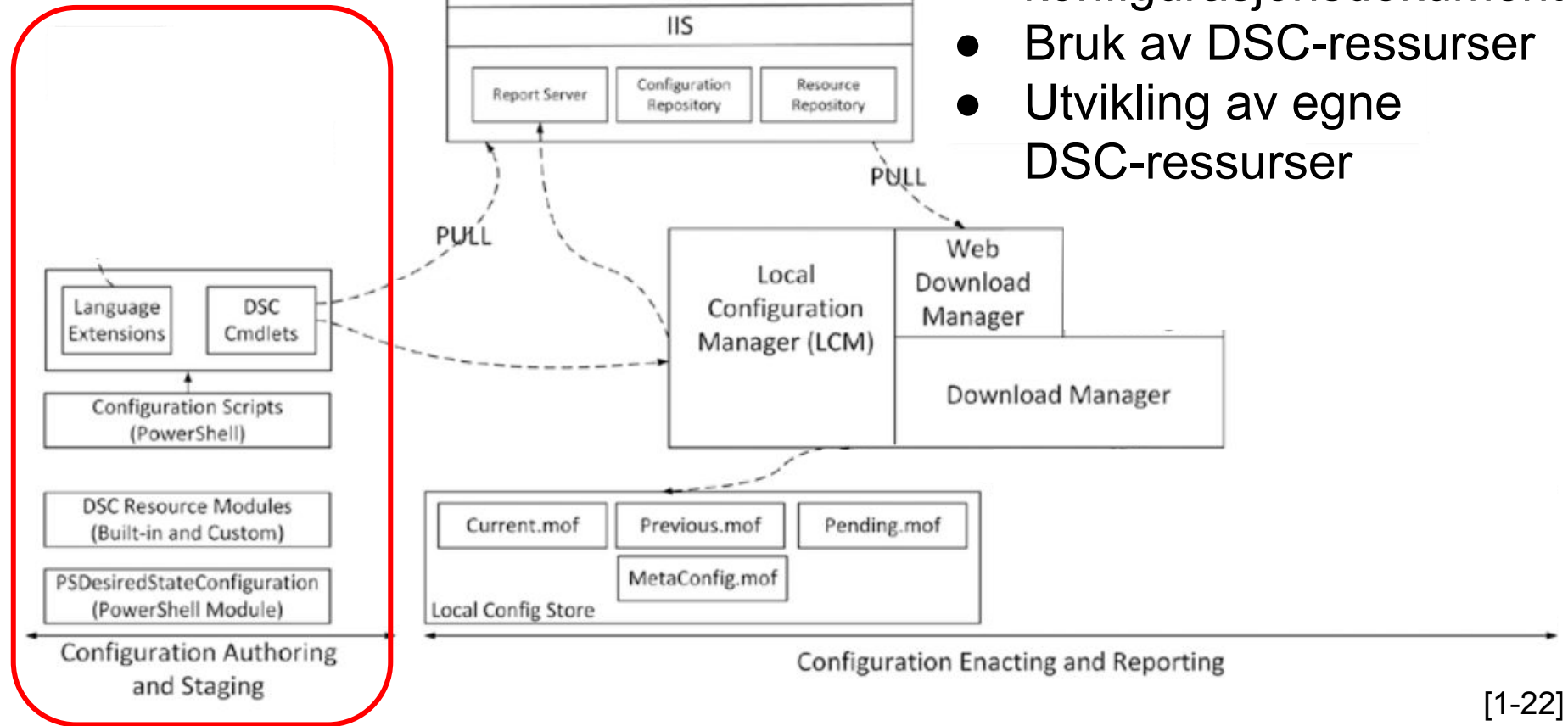


Azure DSC



Utviklingsfasen

Utvikling av konfigurasjon



Utvikling av konfigurasjon

- Deklarativ
- Idempotent

```
Configuration ConfigurationExample
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node SessionHost
    {
        WindowsFeature RDS
        {
            Ensure = "Present"
            Name = "RDS-RD-Server"
        }
    }
}
```

Utvikling av konfigurasjon - Konfigurasjonsblokk

```
Configuration ConfigurationExample ← Konfigurasjonsblokk
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node SessionHost
    {
        WindowsFeature RDS
        {
            Ensure = "Present"
            Name = "RDS-RD-Server"
        }
    }
}
```

Utvikling av konfigurasjon - Nodeblokk

```
Configuration ConfigurationExample  ← Konfigurasjonsblokk
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration
    Node SessionHost  ← Nodeblokk
    {
        WindowsFeature RDS
        {
            Ensure = "Present"
            Name = "RDS-RD-Server"
        }
    }
}
```

Navn på konfigurasjon: ConfigurationExample.SessionHost

Utvikling av konfigurasjon - Ressursblokk

```
Configuration ConfigurationExample
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node SessionHost
    {
        WindowsFeature RDS
        {
            Ensure = "Present"
            Name = "RDS-RD-Server"
        }
    }
}
```

← Konfigurasjonsblokk

← Nodeblokk

← Ressursblokk

Navn på konfigurasjon: ConfigurationExample.SessionHost

Utvikling av konfigurasjon - Ressursblokk

```
Configuration ConfigurationExample
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node SessionHost
    {
        WindowsFeature RDS
        {
            Ensure = "Present"
            Name = "RDS-RD-Server"
        }
    }
}
```

← Konfigurasjonsblokk

← Nodeblokk

← Ressursblokk

← Navn på Windows Feature

Navn på konfigurasjon: ConfigurationExample.SessionHost

Utvikling av konfigurasjon - Ressursblokk

```
Configuration ConfigurationExample ← Konfigurasjonsblokk
{
  Import-DscResource -ModuleName PSDesiredStateConfiguration
  Node SessionHost ← Nodeblokk
  {
    WindowsFeature RDS ← Ressursblokk
    {
      Ensure = "Present" ← Ønsket tilstand
      Name = "RDS-RD-Server" ← Navn på Windows Feature
    }
  }
}
```

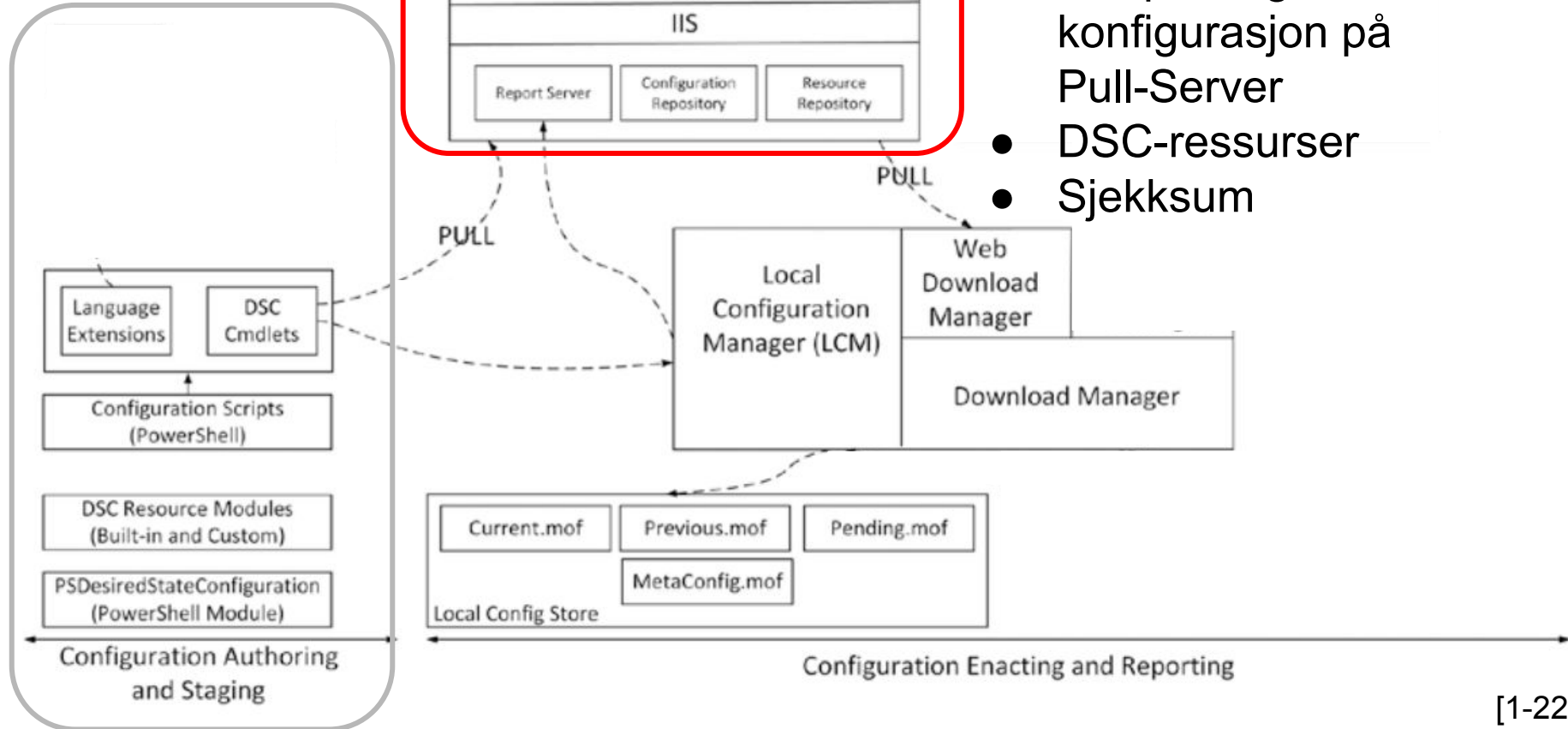
Navn på konfigurasjon: ConfigurationExample.SessionHost

Utvikling av konfigurasjon - Kort om DSC-ressurser

- Get
 - Returnerer egenskapene til ressursen
 - Sjekker om RD-RDS-Server er tilstede og returnerer navn + present / absent
- Test
 - Kontrollerer om maskinen er i den ønskede tilstanden
 - Returnerer True / False
- Set
 - Konfigurerer maskinen slik at den settes i den ønskede tilstanden

Staging

Utvikling av konfigurasjon



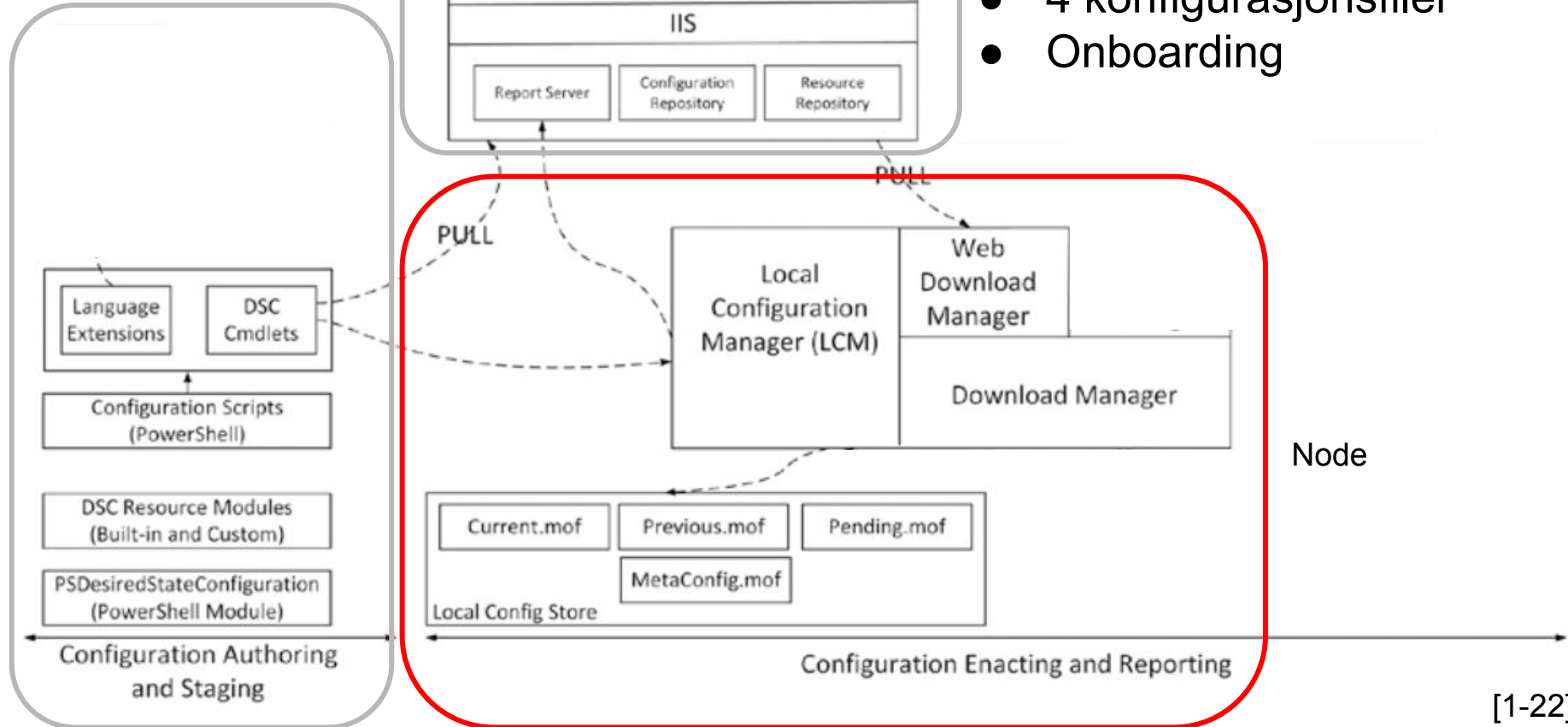
- Overføring til og kompilering av konfigurasjon på Pull-Server
- DSC-ressurser
- Sjekksum

Demo

Opplastning og kompilering av konfigurasjon

LCM - Enact

Utvikling av konfigurasjon

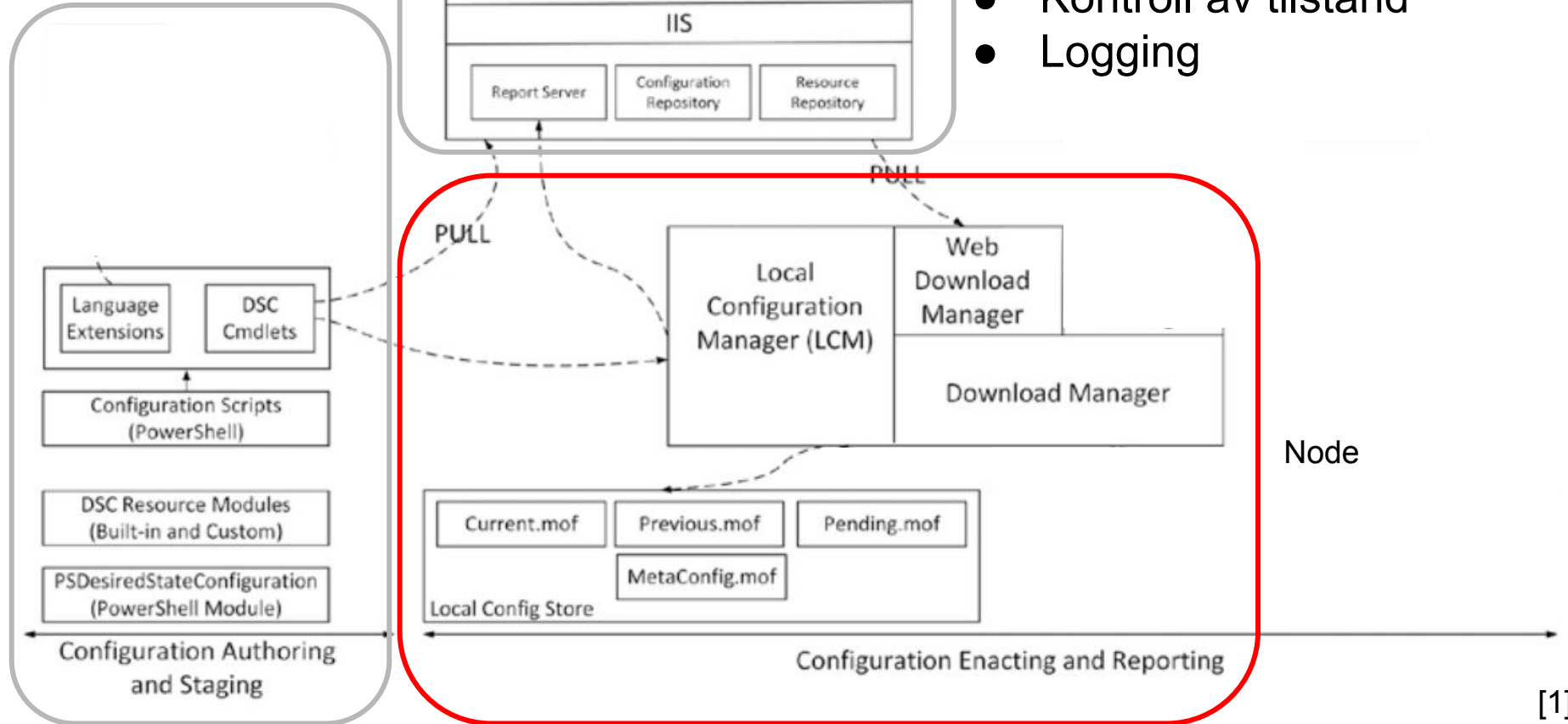


Demo

Konfigurasjon av LCM og onboarding

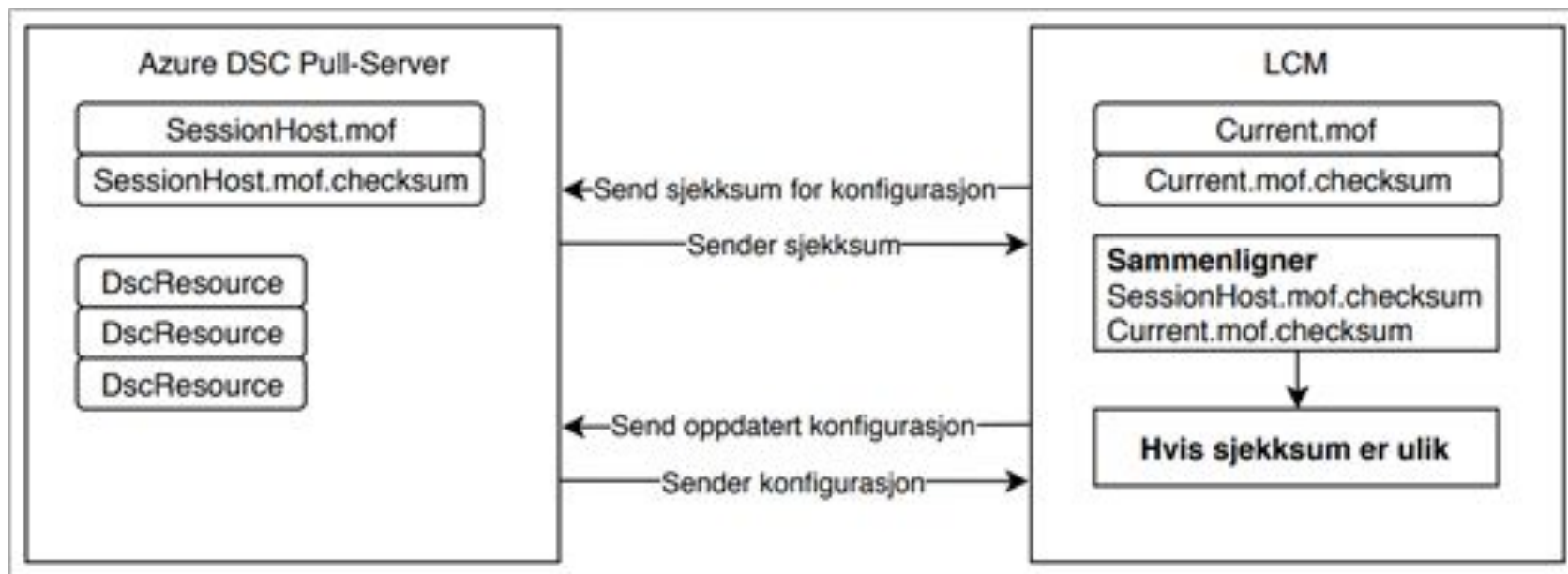
LCM - Enact

Utvikling av konfigurasjon



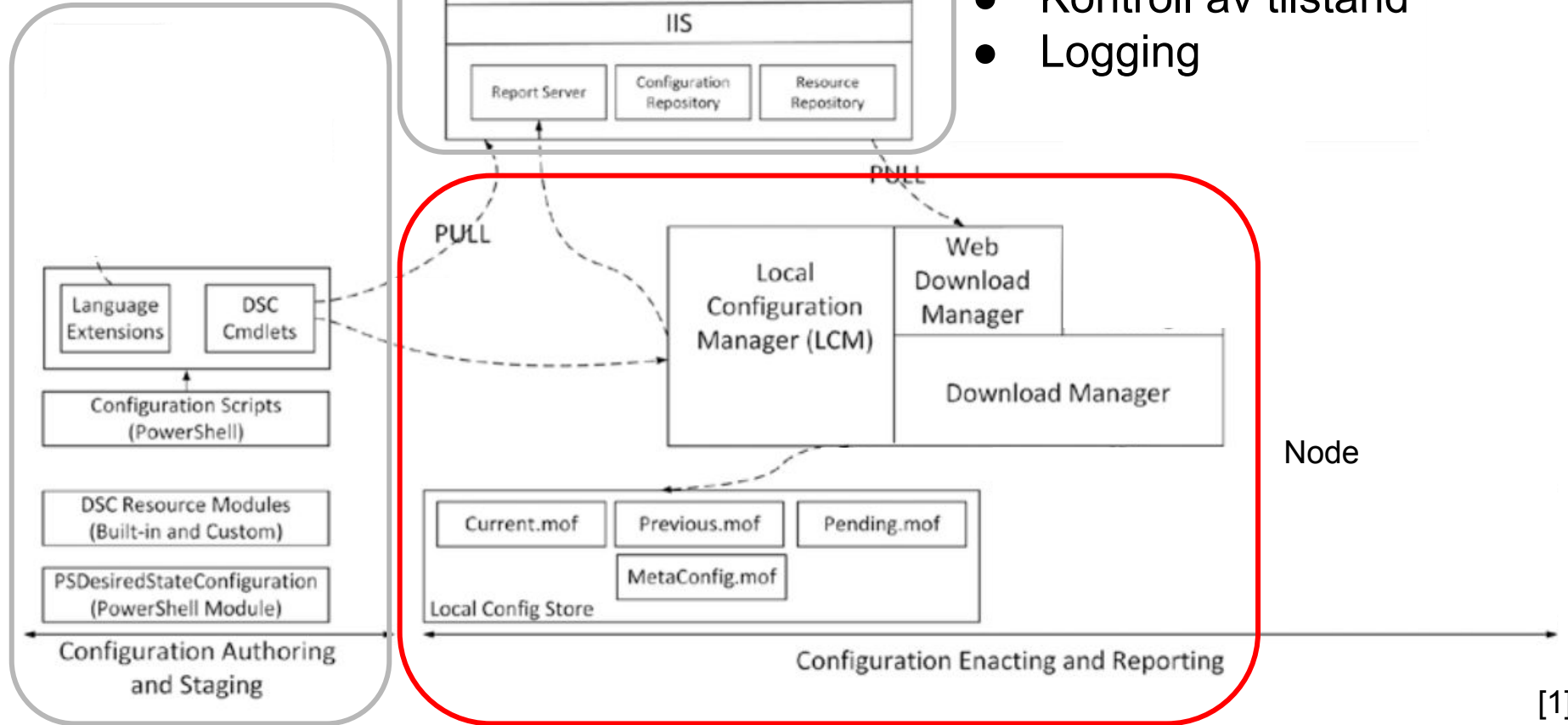
Henting av konfigurasjon

- “RefreshFrequencyMins” (Hvert 30. minutt)



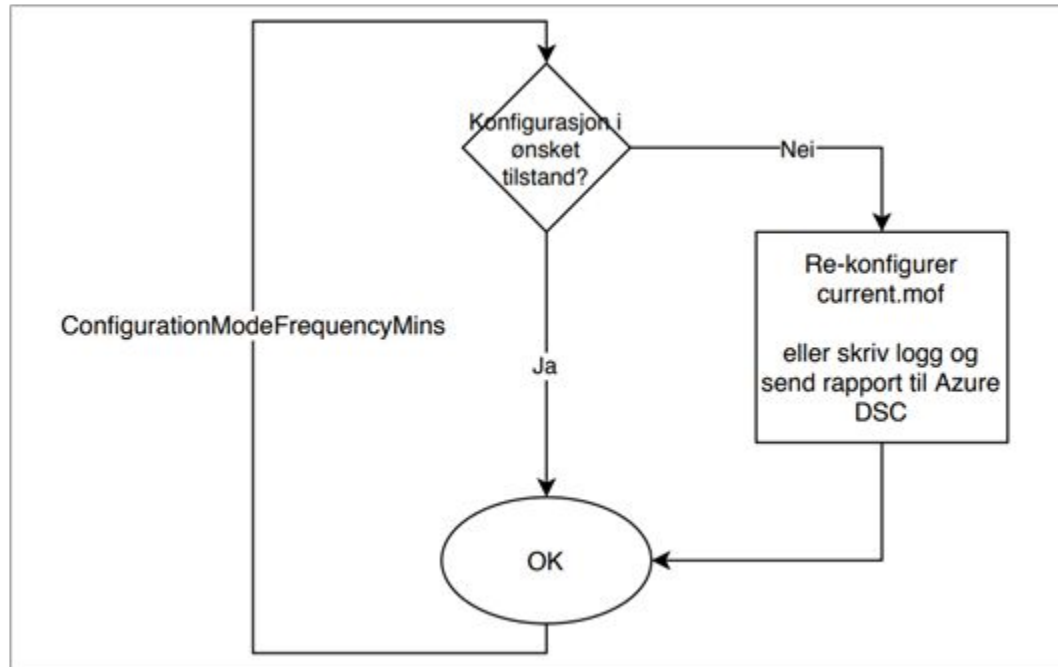
LCM - Enact

Utvikling av konfigurasjon



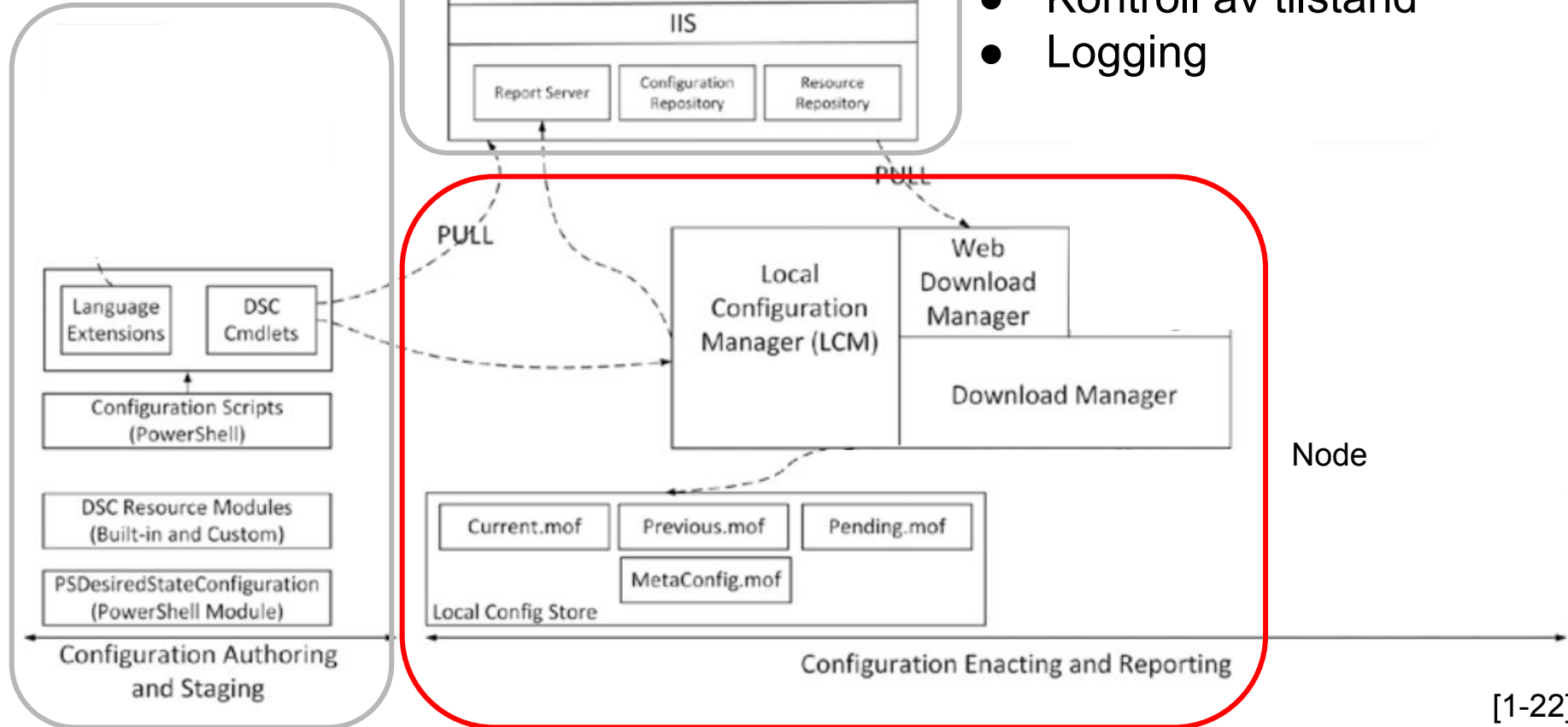
Kontroll av tilstand

- “ConfigurationModeFrequencyMins” (Hvert 15. minutt)



LCM - Reporting

Utvikling av konfigurasjon



Demo

Logging og rapportering

Konklusjon

- Enkelt å komme igang - Krever ikke veldig høy kompetanse
 - Portalen gjør det enkelt
- Erstatte ikke MDT og GPO... Men passer godt inn i mellom?
- Bruksområde?
 - Fungerer på alt som kan løses med PowerShell
- Behov for To-faktor-autentisering i Azure
- Problemer med DSC-ressurs for å konfigurere Session Host
 - WinRM trafikk er sperret
 - Konfigurasjon vs administrasjon
- Opprydding hvis konfigurasjon fjernes

Spørsmål?

Kilder

- [1] Chaganti, R. Pro PowerShell Desired State Configuration: An In-Depth Guide to Windows PowerShell DSC 2nd ed., Berkeley, CA. 2018