

Standardize your Cloud using Feature Store

An AWS Community Day Presentation

Vaibhav Shah | Jeroen Nijssen





AGENDA

data - date

titolo - title

autore - author

premi letterari - awards

editore - publisher

libreria - bookstore

biblioteca - library

suggerito da - suggested by

commenti - comments

n°

data - date

titolo - title

autore - author

premi letterari - awards

editore - publisher

libreria - bookstore

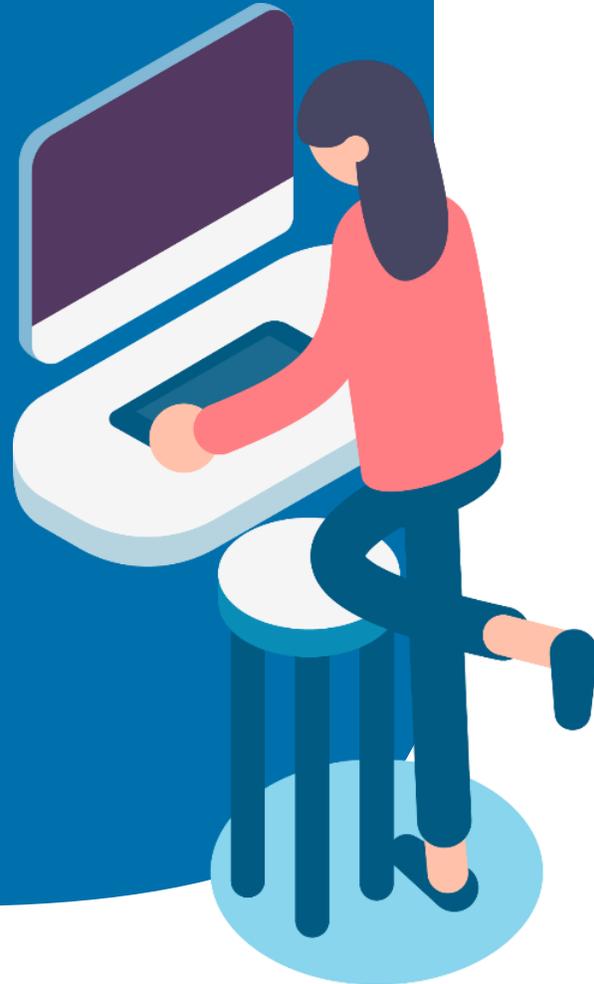
biblioteca - library

suggerito da - suggested by

commenti

prestato a/da

Agenda



- Introduction
- The Feature Store
- Deep dive into Patching feature
- Lessons learned
- Q/A

A close-up photograph of two hands shaking in a firm grip. The hands are positioned horizontally across the frame, with the left hand on the left and the right hand on the right. The skin is a warm, light brown tone. The background is a plain, light gray. A semi-transparent dark gray rectangular box is overlaid across the center of the hands, containing the word "INTRODUCTION" in white, bold, uppercase letters.

INTRODUCTION

WHO AM I?



Name:

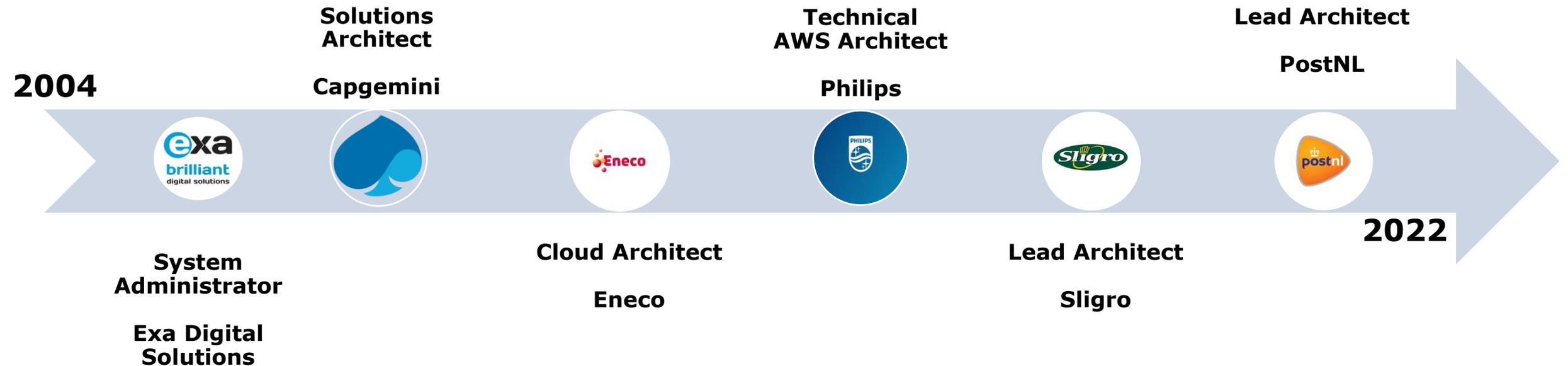
Vaibhav Shah

vaibhav.shah@sogeti.com

Function:

Lead Architect - AWS

Career journey:



WHO AM I?

Name: Jeroen Nijssen
jeroen.nijssen@sogeti.com
Function: Cloud Consultant



Career journey:

Research
Intern

NFI



Team Lead AWS
Services

Sogeti



Cloud
Consultant

Sligro



2022

2015

Technical
Engineer

Van Oord



Lead AWS
Consultant

Philips



Cloud
Consultant

PostNL



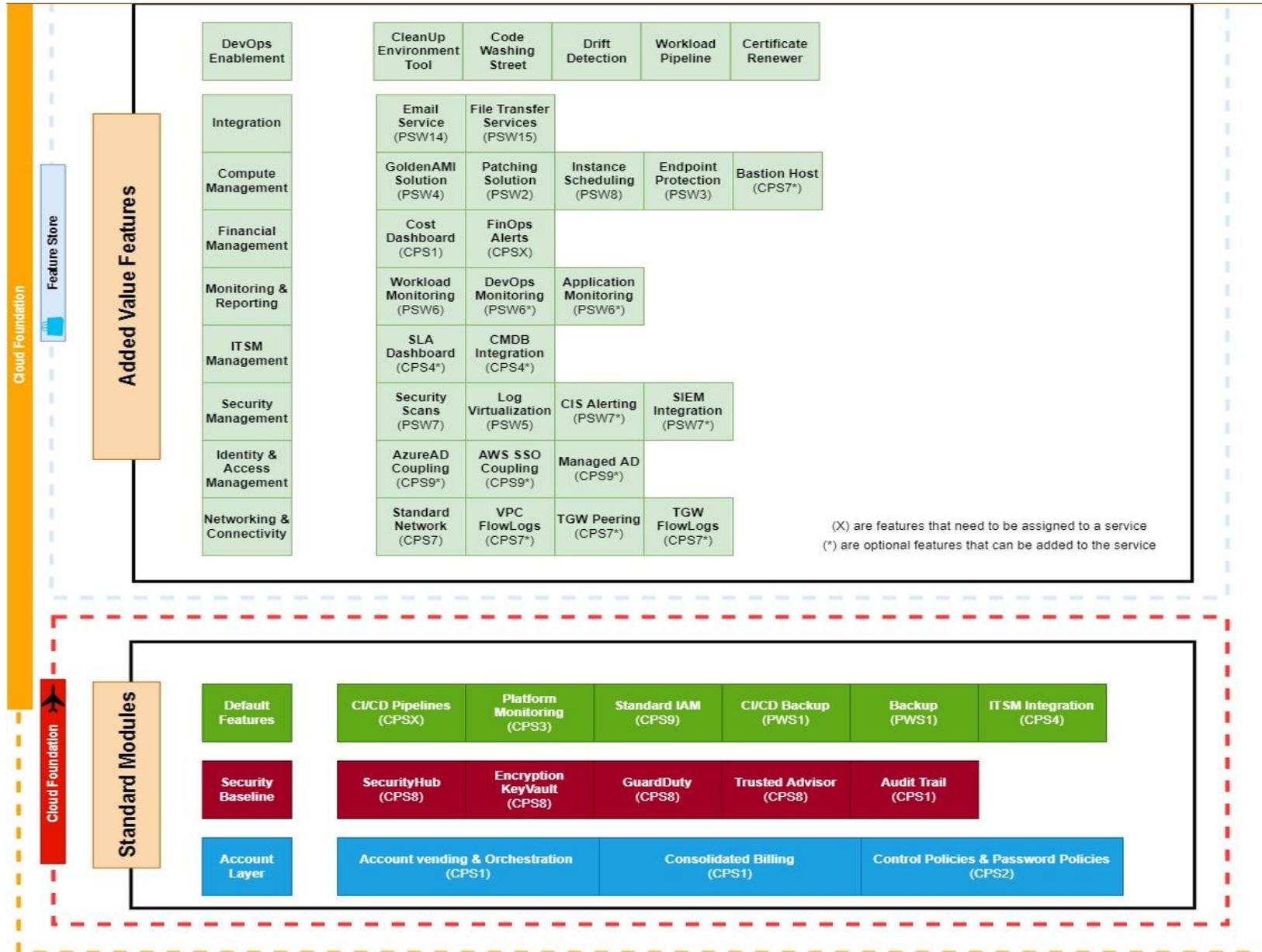


THE FEATURE STORE

4 Reasons why we built Feature Store

- To enable and accelerate our platform and developer teams on cloud.
- To provide paradigm for our small, medium and enterprise customers to be able to land applications safely.
- Deploy well built operational modules and expedite cloud migrations based on standardized patterns.
- To conform to security and architectural best practices

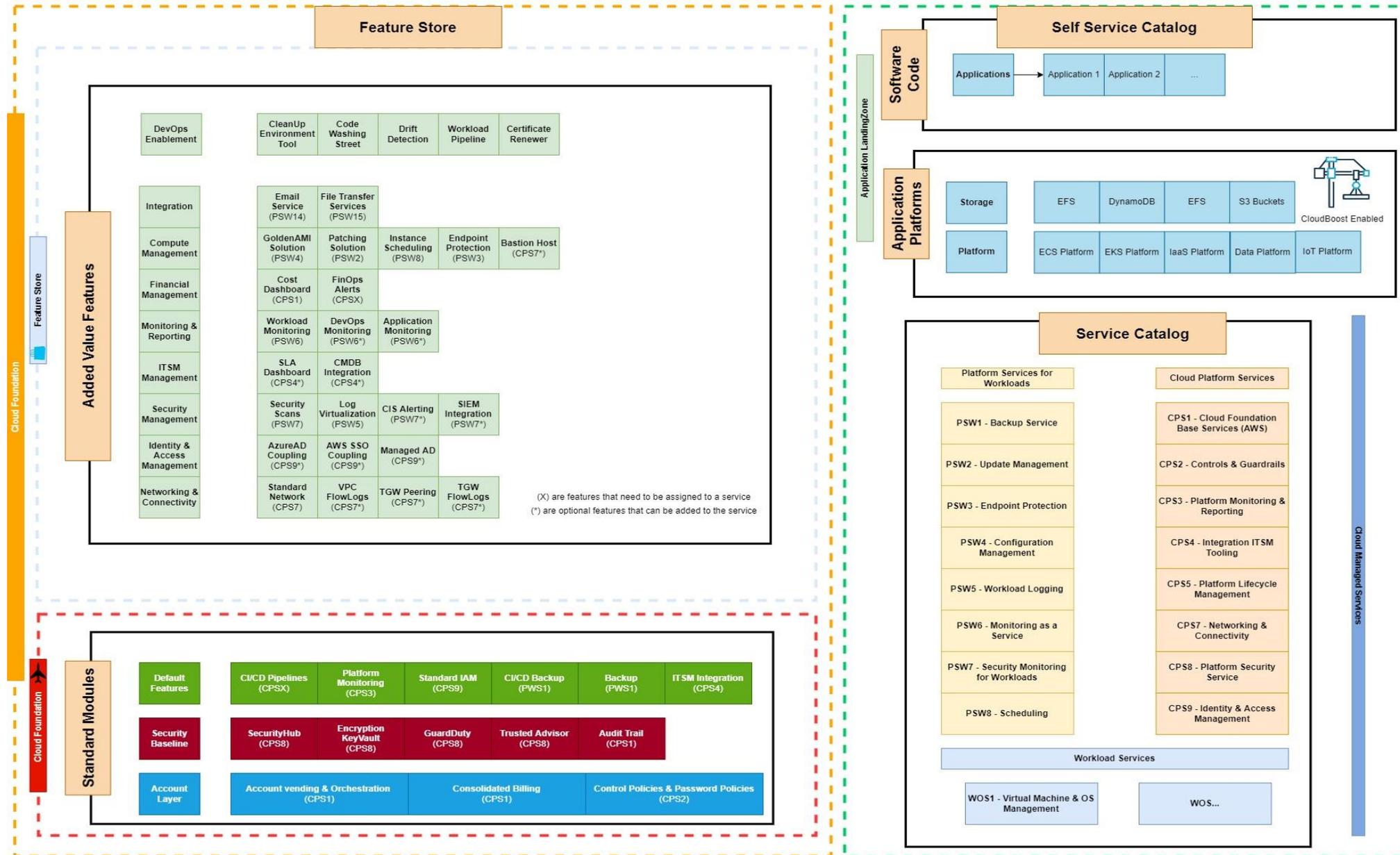
The Feature store

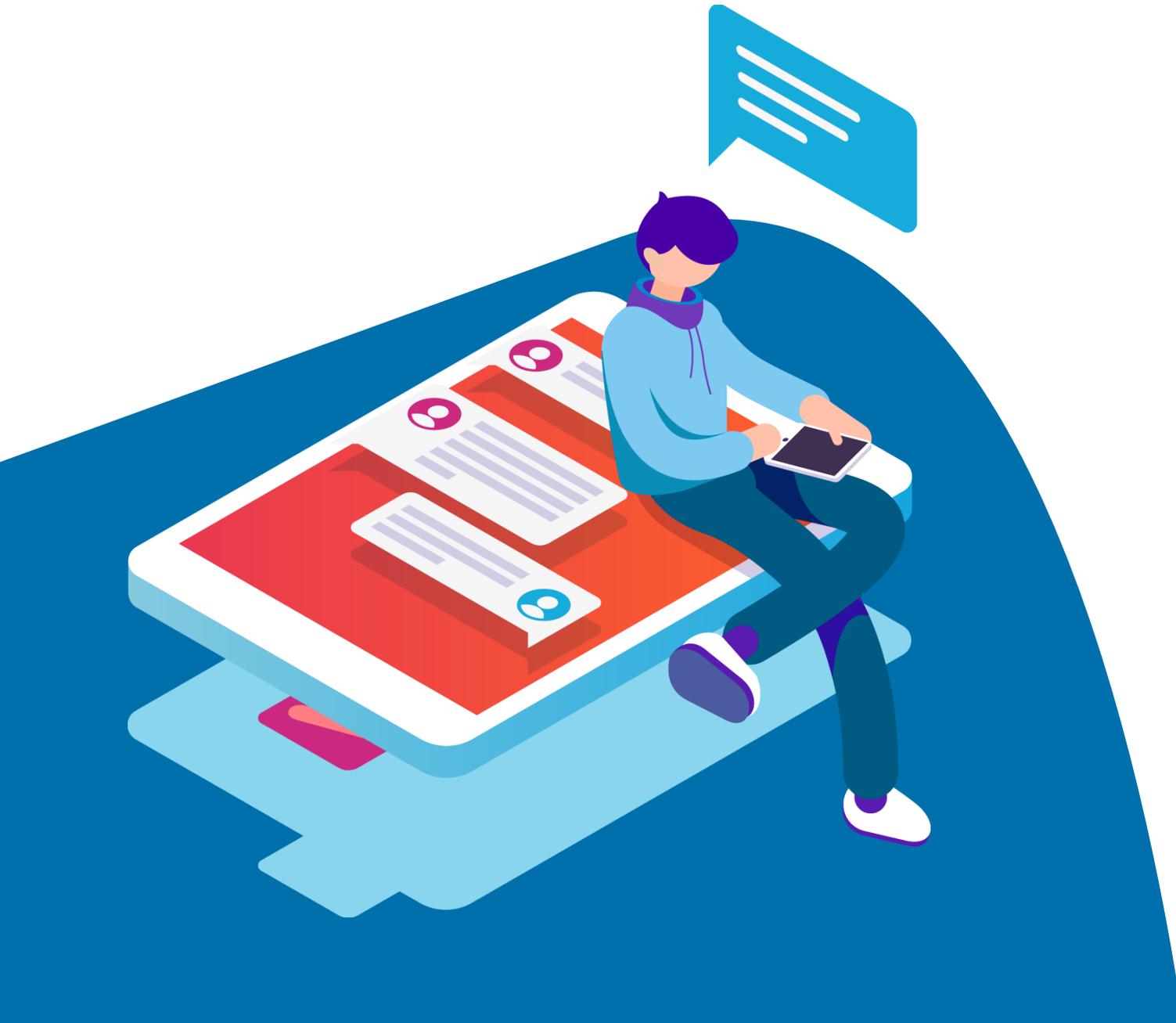


Features

Service Constructs	Foundation Features	Value Added Features
<p>Ways how we deliver the features towards the customers</p>	<p>Give the application a place to land within the landscape of the customer</p>	<p>Add value to the landing place.</p>
<ul style="list-style-type: none"> - Service Catalog - Self Service (Catalog) 	<ul style="list-style-type: none"> - Baseline - Policies - Monitoring - ITSM Features - Lifecycle - Certificate & Key - Networking & Connectivity - Security - Operational IAM - Platform CI/CD - Incident Reporting 	<ul style="list-style-type: none"> - Backup - Patching - Endpoint Protection - Monitoring (workload) - Scheduling - Compute Optimizer - Cloud Economics - AMI Lifecycle - CIS Alerting - SIEM Integration - SMTP Services - FTP - Drift detection - Workload pipelines

Cloud Foundation





Statements

“Petje op petje af”

“I Don't use Control Tower, because I know better what my company/clients need.”

“There is no added value in security products.”

“Cloud Foundations should be modular.”

“Infra(structure) management is dead.”



PATCHING FEATURE

Deep Dive – Feature | Patching

CVE-2022-30190, again! This must be the number 9 thousand for this year.

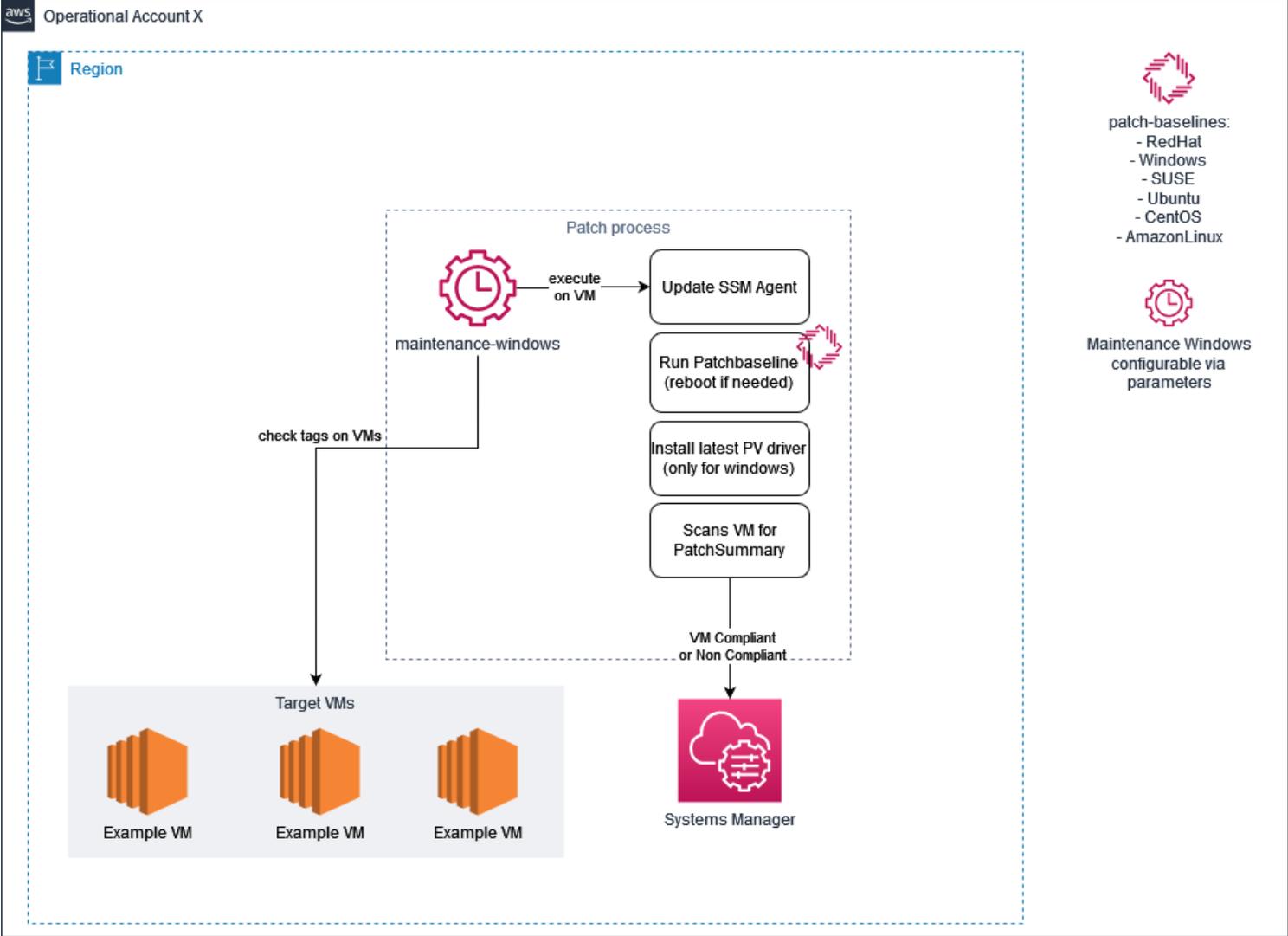
Within our Value Delivery Center, we take care of customers' needs daily. We delivery 24/7 support on there most critical systems and make sure that they functional workflows are not interrupted.

One of those needs, is obviously patch management. Within that need we always advice customers to stay as close to the hyperscale's are possible. Within AWS that means that we make use of the features of AWS Systems Manager

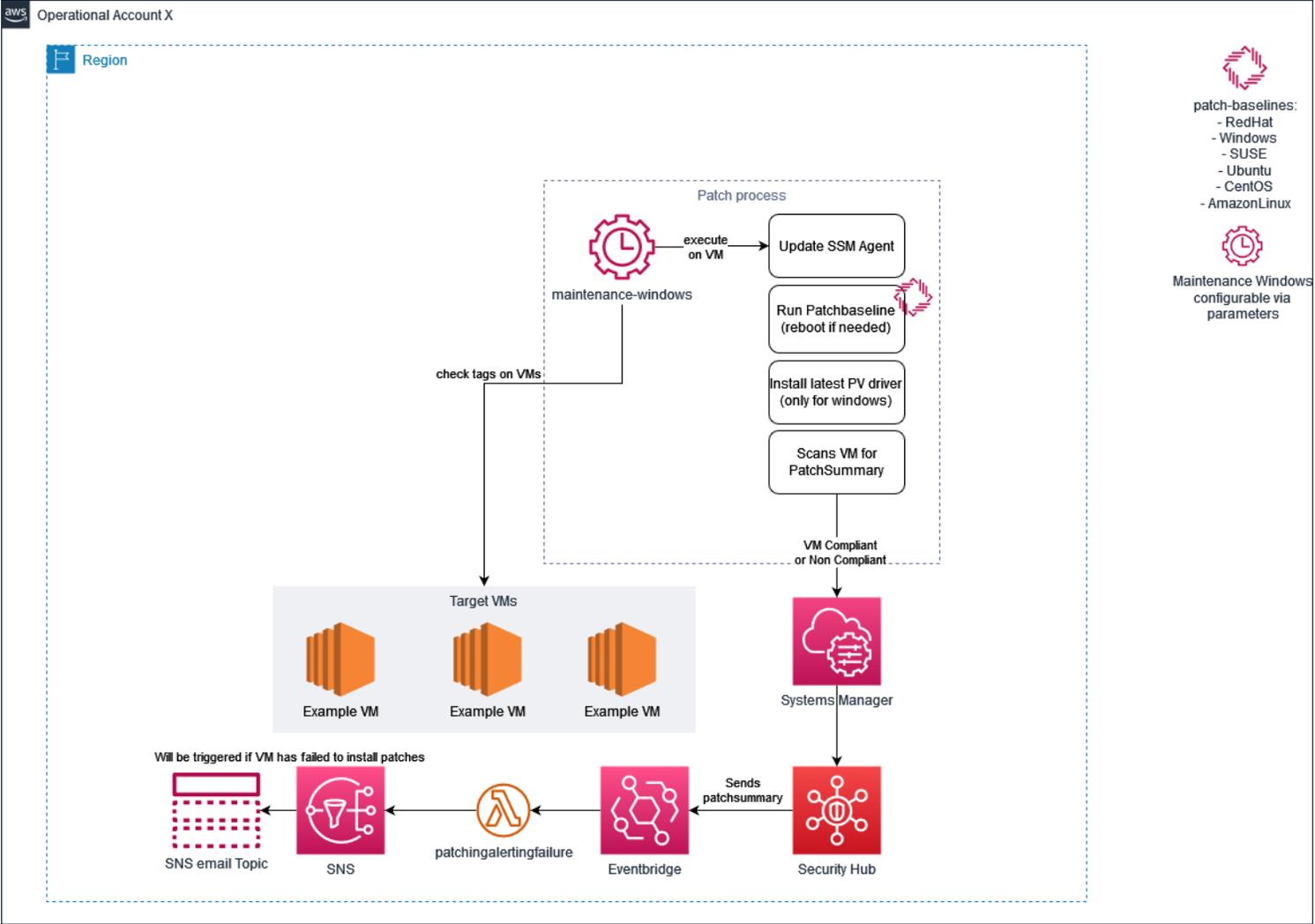
But how does that look like...

“ Microsoft Windows Support Diagnostic Tool (MSDT) Remote Code Execution Vulnerability
CVE-2022-30190

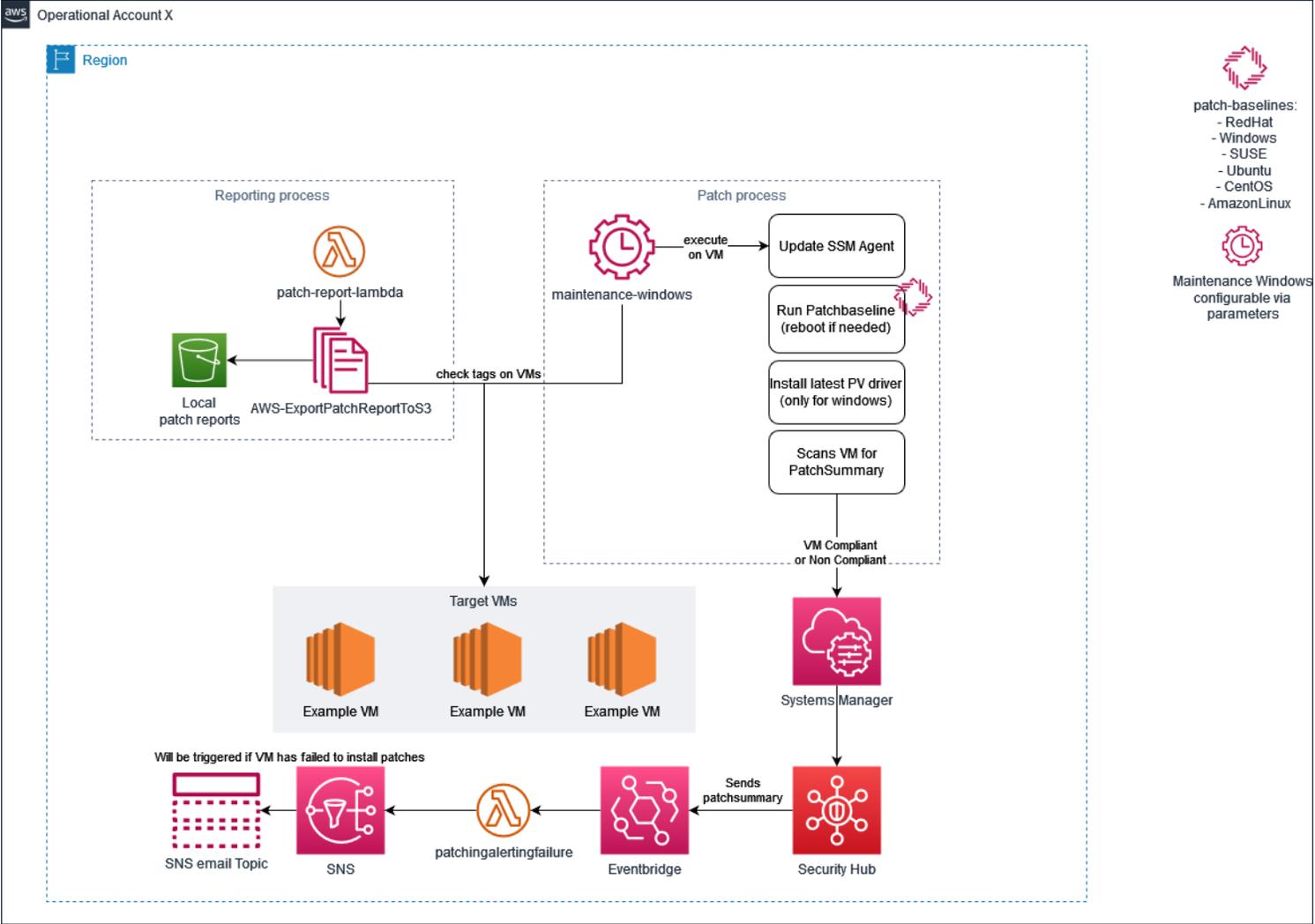
Patching is easy



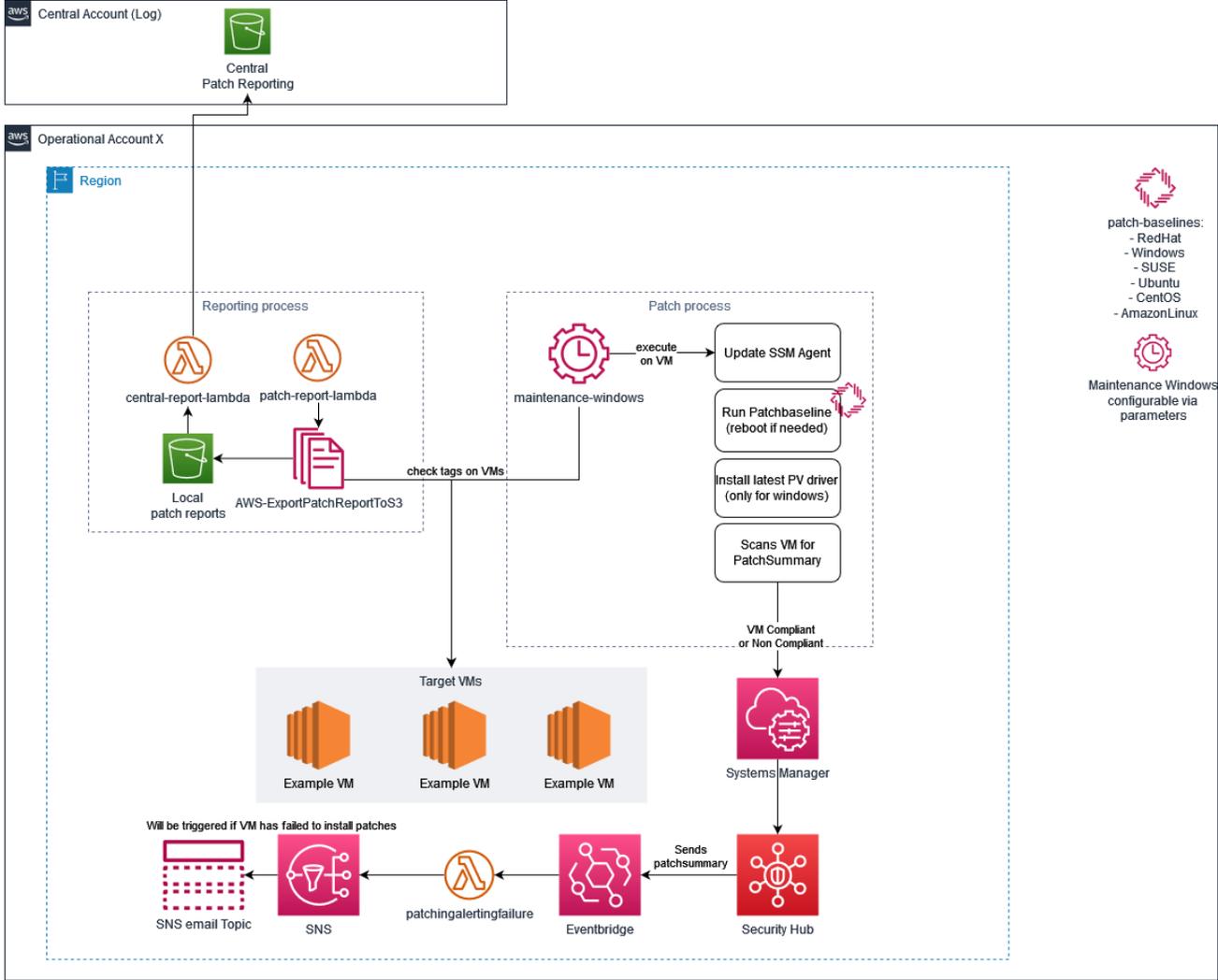
Fair, we are patching



Can we just focus on the main objective



Okay, there you have it.



Now it's easy again.

```
1 module "Patching" {
2   source = "../..//SSMPatching" #Internal testing
3
4   awsIz          = "true"
5   costcenter     = "Generic"
6   costcenter_short = "gen"
7   customer_short = "example"
8   environment    = "p"
9
10  patch_tag      = "Patch Group"
11  schedule_group1 = "cron(00 0 ? * 1#2 *)" #second sunday of every month
12  schedule_group1_duration_hours = 3
13  schedule_group1_offset_days    = 0
14  schedule_group2 = "cron(00 0 ? * 1#4 *)" #fourth sunday of every month
15  schedule_group2_duration_hours = 3
16  schedule_group2_offset_days    = 0
17  schedule_timezone              = "Europe/Amsterdam"
18
19  linux_approval_after_days = 1
20  linux_maintenance_task_settings = {
21    "max_concurrency" : "25",
22    "max_errors"      : "12"
23  }
24
25  windows_approval_after_days = 2
26  windows_maintenance_task_settings = {
27    "max_concurrency" : "100",
28    "max_errors"      : "90"
29  }
30
31  rejected_patches_centos      = []
32  rejected_patches_linux      = []
33  rejected_patches_redhatenterprise = []
34  rejected_patches_suse       = []
35  rejected_patches_ubuntu     = []
36  rejected_patches_windows    = []
37 }
```

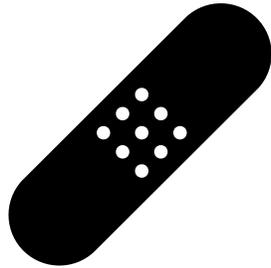
Now it's easy again.

When using only defaults

```
1 module "Patching" {
2   source = "../..//SSMPatching" #Internal testing
3
4   costcenter      = "Generic"
5   costcenter_short = "gen"
6   customer_short  = "example"
7   environment     = "p"
8 }
```

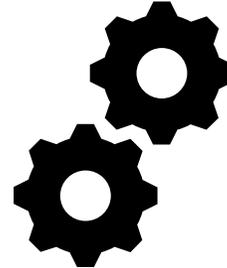
```
1 module "Patching" {
2   source = "../..//SSMPatching" #Internal testing
3
4   awsiz          = "true"
5   costcenter     = "Generic"
6   costcenter_short = "gen"
7   customer_short = "example"
8   environment    = "p"
9
10  patch_tag      = "Patch Group"
11  schedule_group1 = "cron(00 0 ? * 1#2 *)" #second sunday of evey month
12  schedule_group1_duration_hours = 3
13  schedule_group1_offset_days    = 0
14  schedule_group2 = "cron(00 0 ? * 1#4 *)" #fourth sunday of evey month
15  schedule_group2_duration_hours = 3
16  schedule_group2_offset_days    = 0
17  schedule_timezone = "Europe/Amsterdam"
18
19  linux_approval_after_days = 1
20  linux_maintenance_task_settings = {
21    "max_concurrency" : "25",
22    "max_errors"      : "12"
23  }
24
25  windows_approval_after_days = 2
26  windows_maintenance_task_settings = {
27    "max_concurrency" : "100",
28    "max_errors"      : "90"
29  }
30
31  rejected_patches_centos      = []
32  rejected_patches_linux      = []
33  rejected_patches_redhatenterprise = []
34  rejected_patches_suse       = []
35  rejected_patches_ubuntu     = []
36  rejected_patches_windows    = []
37 }
```

The configuration



patch-baselines:

- RedHat
- Windows
- SUSE
- Ubuntu
- CentOS
- AmazonLinux



Maintenance Windows
configurable via
parameters



Scheduling the Patches

Bringing back to the community

- Feature requests on AWS
 - Installation Time
- Involvement of Microsoft on Patches
- Good Practices towards operational teams



LESSONS LEARNED

Lesson Learned

- Historical agreements
- People are afraid of change, the unknown
- Staying close to the hyperscaler
- Build guardrails not roadblocks





WHAT

QUESTIONS ?

About Sogeti

Part of the Capgemini Group, Sogeti operates in more than 100 locations globally. Working closely with clients and partners to take full advantage of the opportunities of technology, Sogeti combines agility and speed of implementation to tailor innovative future-focused solutions in Digital Assurance and Testing, Cloud and Cybersecurity, all fueled by AI and automation. With its hands-on 'value in the making' approach and passion for technology, Sogeti helps organizations implement their digital journeys at speed.

A global leader in consulting, digital transformation, technology and engineering services, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year+ heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. Today, it is a multicultural company of 270,000 team members in almost 50 countries. With Altran, the Group reported 2022 combined revenues of €17billion.

Visit us at
www.sogeti.com



This message contains information that may be privileged or confidential and is the property of the Capgemini Group.

Copyright© 2022 Sogeti. All rights reserved.