ATLASSIAN

The content described herein is intended to outline our general product direction for informational purposes only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described herein remain at the sole discretion of Atlassian and is subject to change.

www.cprime.com | 877.800.5221 | +44 (0) 203 811 0424

A Jira enterprise success story: Applying CI/CD solutions

cprime | T-Mobile

www.cprime.com | 877.800.5221 | +44 (0) 203 811 0424

Agenda

01	**Overview** • Background (past > present > future)
02	**Problem statement** • Problem statement (what > so what)
03	**Solution** •Architecture overview •DevOps automations •Examples, demo, metrics and data, testimonials
04	**The future and its benefits** •Benefits (past > present > future)
05	**References**

cprime | T-Mobile

Vijay Byndoor

Principal Architect

T-Mobile

Vijay is a Solutions Architect with over 19 years of experience within multiple industries in Application Architecture, Enterprise Architecture, Analytics, DWH Design, Web Application Design and Development, DevOps, Cloud Architecture and Development.

Mario Vaccari

Atlassian Solution Architect

Cprime

Mario has an extensive software development and service background, with an eye towards operations flow.

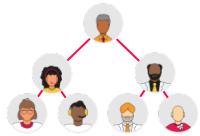
cprime | T-Mobile

Overview

Applying CI/CD solutions to make administering Jira an enterprise success story

This session, co-presented by Cprime and T-Mobile, will look at how the communications leader has applied CI/CD methodologies to control and distribute complex Jira customizations across their organization spanning more than 30,000 users and thousands of departments.

Who is this session for?



- **Large scale:** Teams managing large-scale Atlassian Data Center implementations

- **Multiple customizations:** Implementations requiring multiple and ongoing Jira and JSM customizations tailored to different departments

- **CI/CD automation:** Implementations requiring continuous development, testing, and frequent delivery of Atlassian workloads — Jira and JSM provisioning, onboarding, and customizations

- **Shared development:** Implementations requiring shared environments for configurations and customizations of Atlassian workloads

cprime | T-Mobile

Problem statement

Two years ago, T-Mobile had a large, dated, complex, and diverse tool ecosystem. Various departments needed specific customizations, integrations, and alignment, making enterprise Agile DevOps transformation and adoption complex.

1. Jira and JSM payloads: Complexity in implementing, configuring, version controlling, and deploying multiple variants of Jira and JSM customizations frequently across the diverse departments in T-Mobile
    - Jira and JSM customizations involving UI, Jira gadgets, and scripting (SIL, script-runner, JS)
    - Jira and JSM API management

2. Datacenter infrastructure: Complexity in provisioning large Atlassian datacenter environments on demand and with standard configurations
    - Automated environment provisioning
    - Repeatability, standardization, and infrastructure configuration management
    - Environment true-up and on-demand Jira and JSM environments

3. Automated testing: Automated testing for Jira and JSM
    - Testing Jira and JSM customizations
    - Validating business rules and key scenarios in Jira and JSM

3. DevSecOps: Integrating security and compliance in the DevOps pipelines
    - Asset scanning, static security scanning, secrets scanning
    - Vulnerability scanning

cprime | T-Mobile

Solution

How are we solving the problems?

- Show line between old and new tools
- Build DevOps automations for Atlassian Payloads and Infrastructure

What is the goal?

- Build DevOps automations to implement and deliver Atlassian Payloads in an automated, secure, and quality-centric approach
- Build new flows to automate and facilitate development team needs while satisfying compliance and governance needs efficiently
- Build DevOps automations to provision and maintain datacenter infrastructure on AWS
- On-demand Docker provisioning of Jira and JSM environments for shared Jira/JSM development teams
- Automated jobs to true-up nonproduction environments with the production environment

What are the expectations and boundaries for the audience?

- Large organizations requiring diverse and different Atlassian product customizations with standardization
- Large organizations requiring to provision and maintain datacenter Atlassian products on the cloud

Architecture tech stack

T-Mobile determined the optimal way forward was to adopt a unified and next-gen tool stack that would increase flow through well-integrated systems.

Work management	The new tool stack held Jira Software at the center for work management.
Version control, pipelines, & artifacts	GitLab for version control and CI/CD pipelines.
Asset management	Custom asset management checks with ServiceNow integration.
Quality & security	Fortify, Git Guardian, and Sonar were used for quality and security stages.
Test automation	Qtest and Toscan were used for test automation.
Infrastructure-as-code pipelines	Terraform and Ansible targeting AWS was used for (IAC) infrastructure-as-code pipelines.
Vulnerability scanning	Wazuh was used for vulnerability scanning.
Ephemeral environments	Docker and Kubernetes were used for on-demand Jira Dev environments.
Log aggregation	Elastic/ELK Stack were used for log aggregation and real-time user monitoring.
Monitoring and alerting	OpsGenie.

Solution

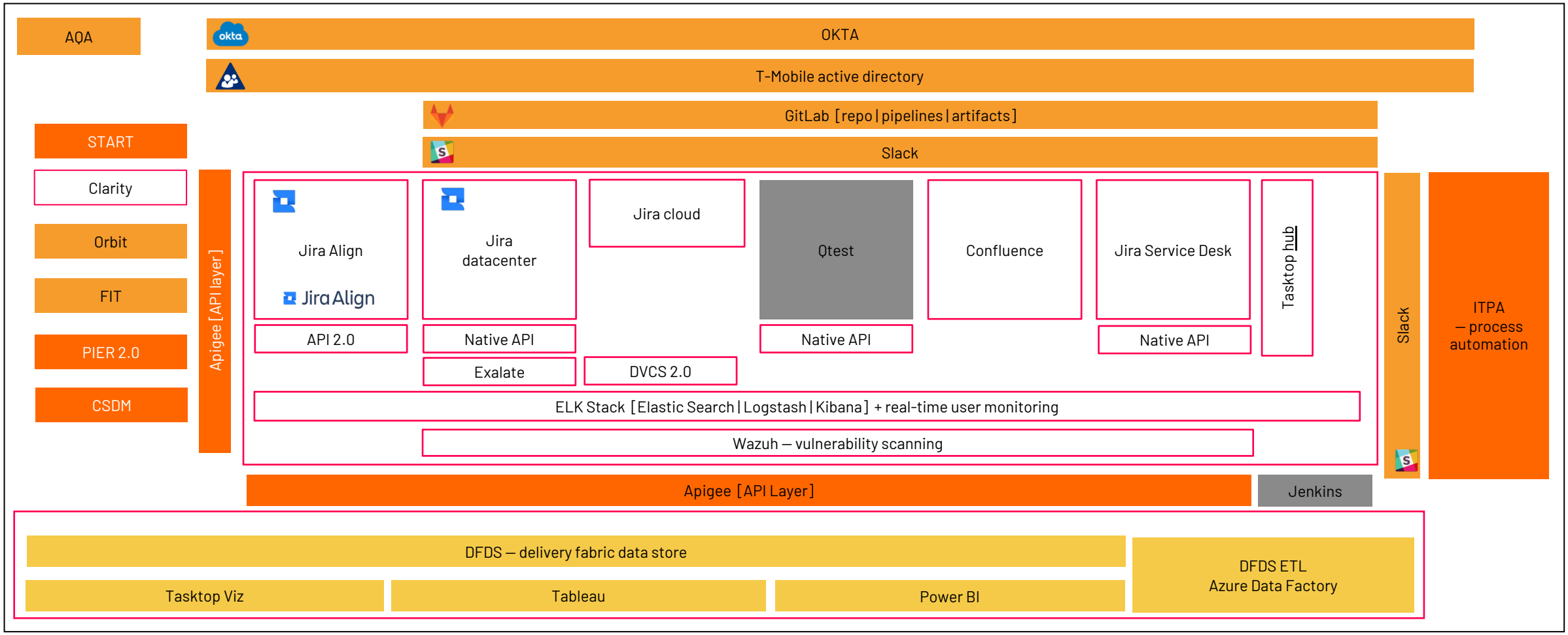
Solution diagrams for each of the below:

- Overall architecture – all the below tools and automations, Elastic/ELK, Wazuh, OpsGenie
- Solution details:
  - **Infrastructure:** Datacenter Infrastructure pipelines using Terraform, Ansible, and Gitlab on AWS
  - **Atlassian Payloads:** CI/CD pipelines for Datacenter Atlassian Payloads using Gitlab
  - **Virtual environments:** Ephemeral Jira environments using Gitlab and AWS
  - **API platform:** CI/D pipelines for Atlassian API platform
  - **Test automation:** Qtest and Tosca Test Automation
  - **Dev-SecOps:** Security and compliance automations in the pipelines
  - **Logs and monitoring:** Data pipelines for log aggregation, monitoring, and alerting
  - **Data pipelines:** Data pipelines for Jira and JSM data warehouse ingestion

PACT – Atlassian architecture overview

Primary components
DFDS components
External components

Intake, capacity & financial mgmt	Roadmapping & portfolio planning	Work & defect management	Test execution & management	Content management/ITSM	Comms., metrics, dashboards



Demo

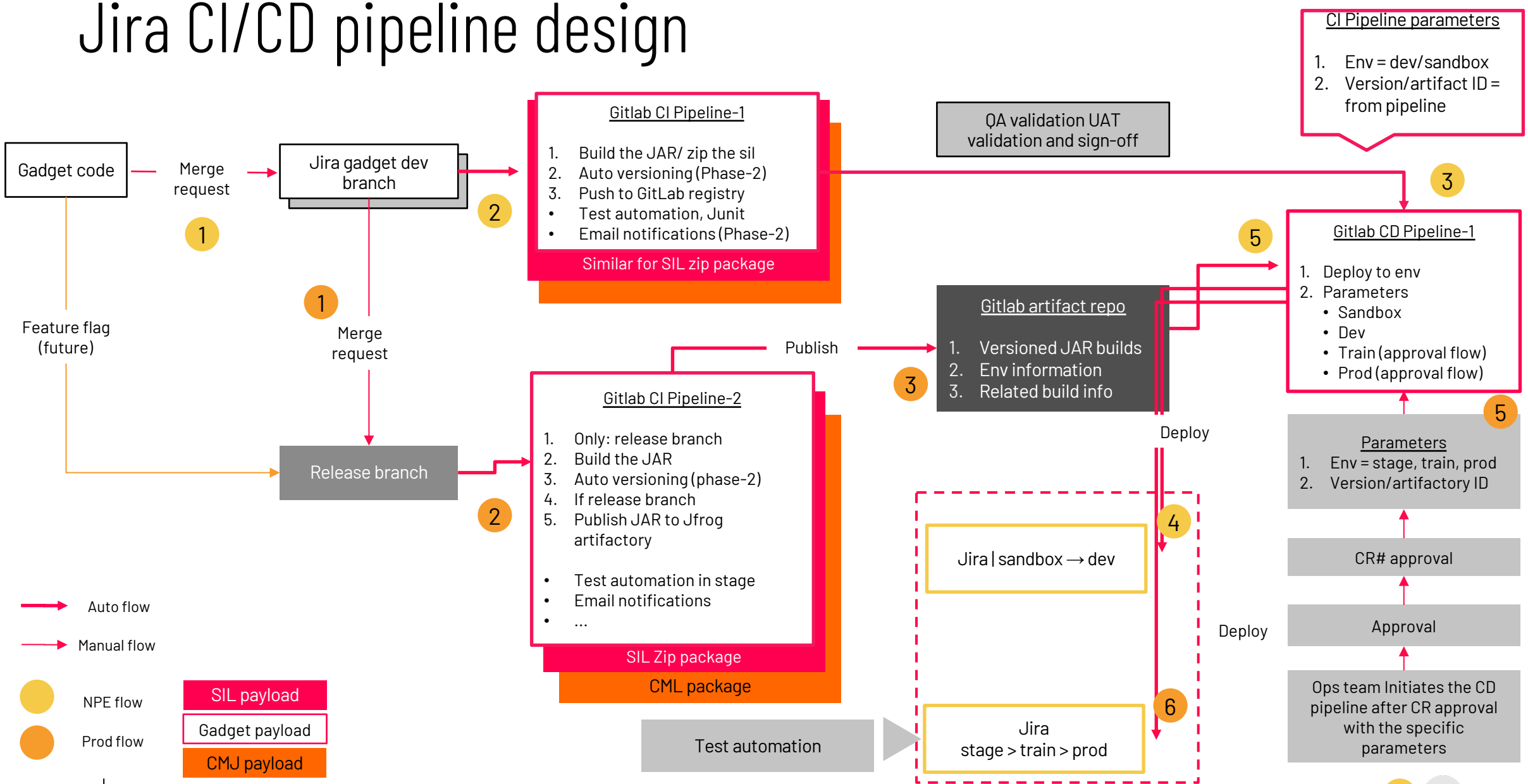
The demo will showcase:

- Datacenter infrastructure pipelines for provisioning on AWS
- Jira and JSM Atlassian Payloads in CI/CD pipelines
- Jira and JSM API platform in CI/CD pipelines
- Gitlab CI/CD pipelines stages
  - Tosca testing stages
  - Security stages
- Ephemeral environments automation

- How we use GitLab and a well-defined change management procedure to speed up Jira customizations, code promotions, and reviews, while raising the standard of value delivered to their customers
- How we use GitLab and Ansible to automate provisioning and maintenance for datacenter servers hosting Jira, JSM, and Confluence on AWS

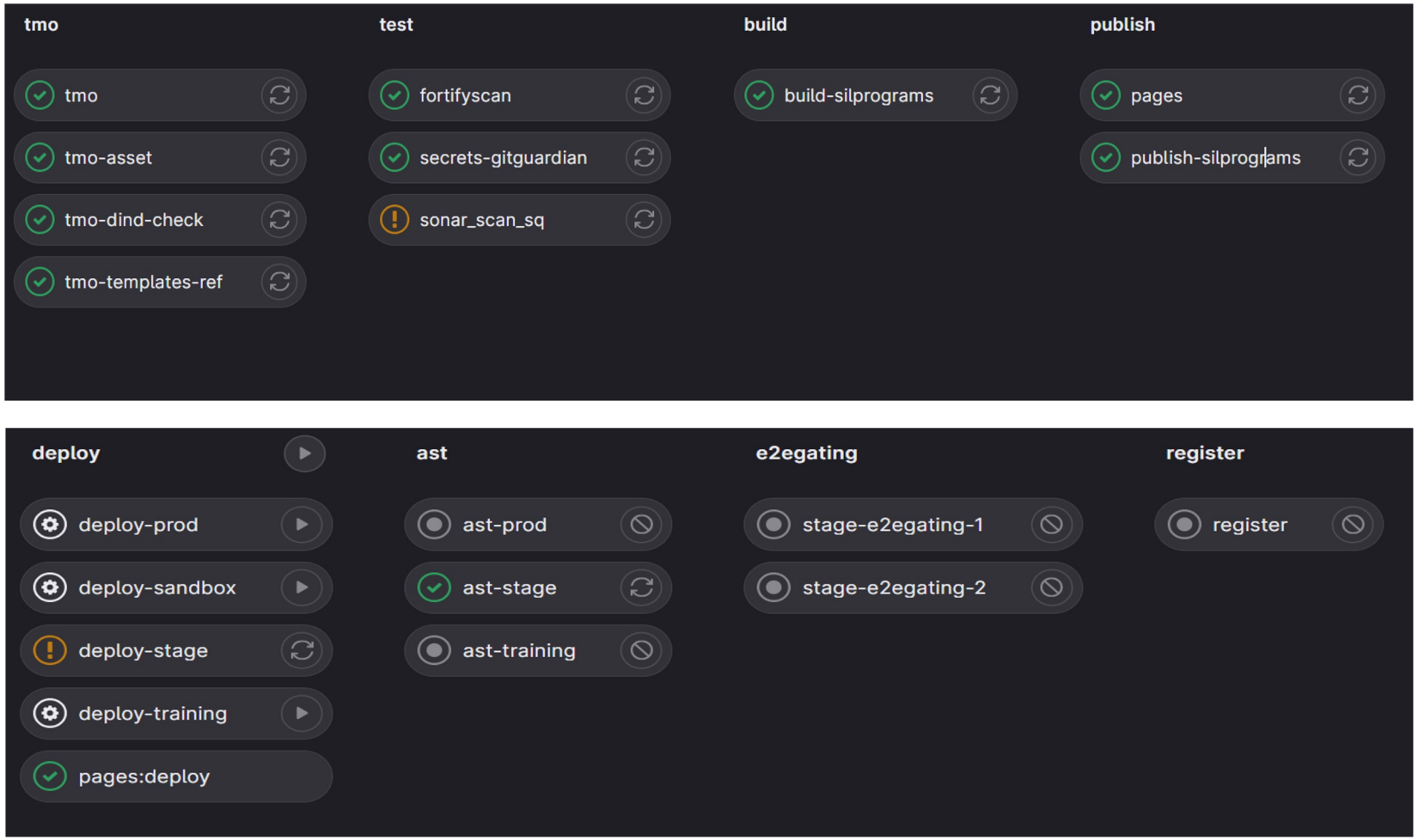
cprime | T-Mobile

Jira CI/CD pipeline design



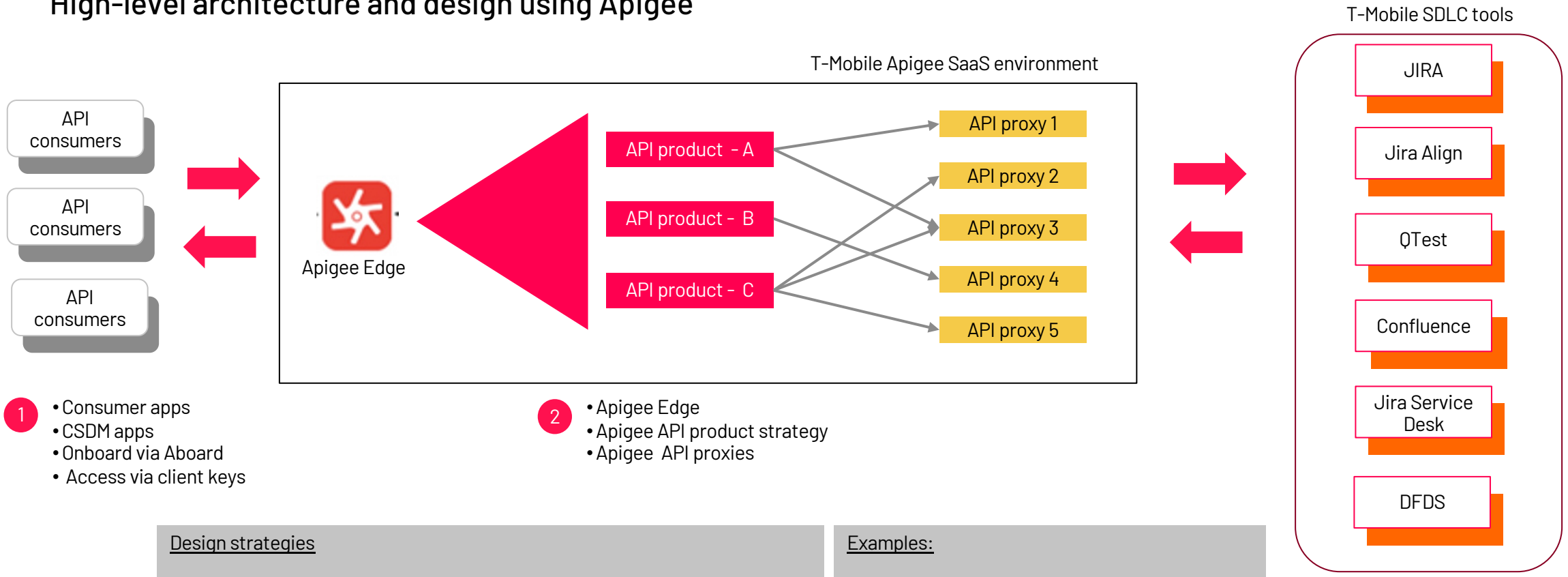
cprime | T-Mobile

GitLab pipeline for Jira-Sil code + security stages



SDLC API platform architecture & design

High-level architecture and design using Apigee



1
- Consumer apps
- CSDM apps
- Onboard via Aboard
- Access via client keys

2
- Apigee Edge
- Apigee API product strategy
- Apigee API proxies

3
- Backend services
- Jira, Jira Align, QTest, etc.

Design strategies

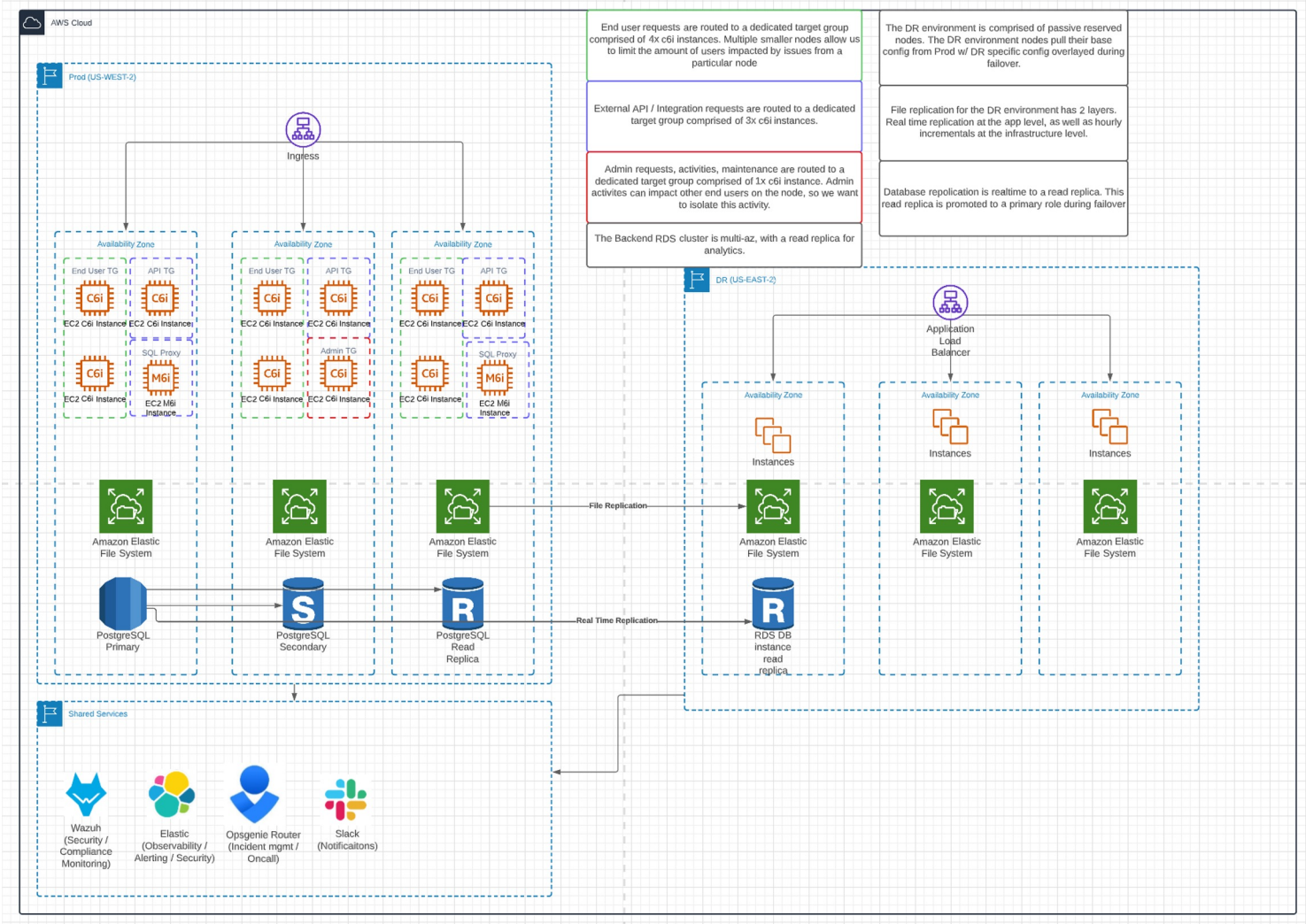
- Product proxy-based model : 1 product : 1 proxy
- Business model : 1 business unit : 1 or more product
- Ownership model :1 development/owner : 1 or more product
- Service plan/tier model : Tier model for different types of APIs

Examples:

Product A : Jira read-only (RO)
Product B: Jira read-write (RO, RW)
Product C: Jira admin (RO, RW, Admin)

cprime | T-Mobile

Infrastructure Architecture



Benefits

- Unifying base processes, training, and integrations across the organization
- Identifying, prioritizing, and addressing
  - Agile principles
  - DevOps automations
  - Security protocols
  - Privacy and risk polices
  - Operational efficiencies through transparency and access to data
- Annual, quarterly, and sprint roadmaps built in partnership with our stakeholders, following the agile approach

- Ability to implement Jira and JSM customization and deliver them using version controlled and automated CI/CD pipelines
- Ability to provision and maintain the Jira, JSM, and Confluence datacenter servers using automated DevOps automations
- Include security and vulnerability scanning in the CI/CD pipelines
- Provide ephemeral Jira and JSM environments for development and testing

NOTE: All functionality delivers data, visibility and control to the operation teams.

cprime | T-Mobile

Benefits

- Benefits (past > present > future)
- Future DevOps automation roadmap
- Enterprise scalability, enterprise agile standardization
- Data and metrics (projects, teams, work items-scale, adoption and growth over time, Git repositories and CICD pipelines

cprime | T-Mobile

Reference Content

R2/D2 mission

Reference Points - 1

1. CI/CD pipelines for continuous delivery of Jira and Jira Service Management changes

T-Mobile's implementation of Atlassian tools is fairly large, with more than 30,000 users spanning multiple departments and parts of the organization.

This has required us to heavily customize the implementation based on specific requirements for different parts of the organization, while at the same time striving to bring consistency and standardization within the organization.

This also brings in multiple standard and custom integrations, both internal and external.

These changes are frequent and complex, driving the need to implement, test, deploy, and monitor automatically at scale and at speed.

This need has driven us to innovate and build CI/CD pipelines to automate our build, testing, and deployment.

This talk will provide a bird's-eye view of the innovative automation we have developed.

- CI/CD pipelines with Gitlab
- Jira and JSD payloads
- Test automation
- Alerting and monitoring

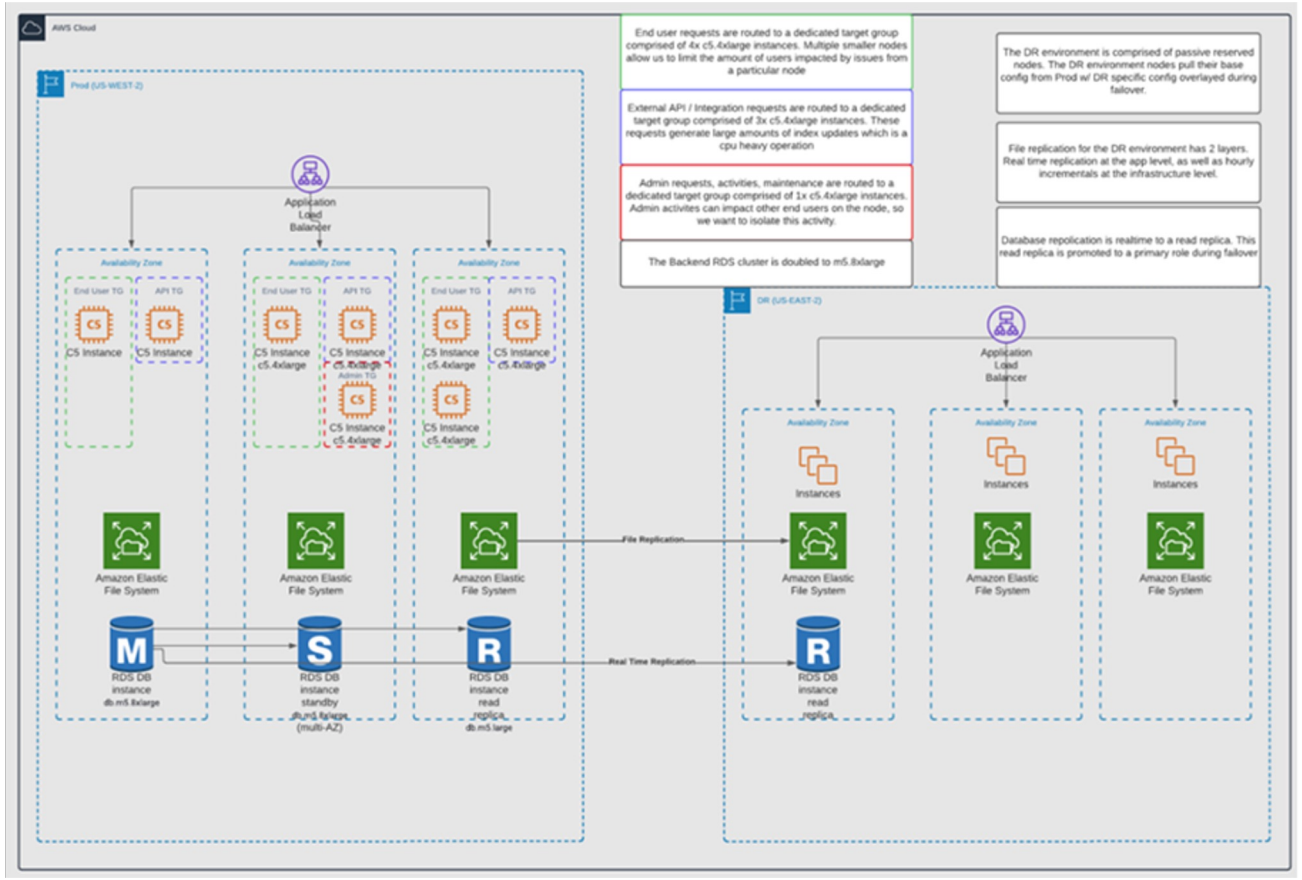
Infrastructure Automation

- Build base AMI's hardened to CIS level-2
- Rebuilt on an interval (weekly or on trigger)
- Use Gitlab as our SCM and CI/CD platform
- All environments stored as code, including application and service configuration
  - Terraform provisions and maintains infra
    - Build base modules for AWS services
    - Then build a 'application/service' module that combines the AWS base module and dependencies into a single module for deployment
  - Ansible utilized for OS/application configuration management
  - All activity sent to Elastic for monitoring and review
  - Subset of activity sent as notifications to Slack

cprime | T-Mobile

Atlassian infrastructure on AWS

- The DR Env is comprised of passive reserved nodes. The DR nodes pull their base config from the PROD environment.
- File replication for DR has two layers:
  - Real-time replication at the app level
  - Hourly replication at the infra level
- DB replication is real-time to the read replica. The replica is promoted to a primary.




Questions?

Thank You

cprime | T-Mobile