MergeBot: Activision's Merge and Resolve Infrastructure

Damianos Gkouzkouris – Senior Tools Engineer

Who am I?

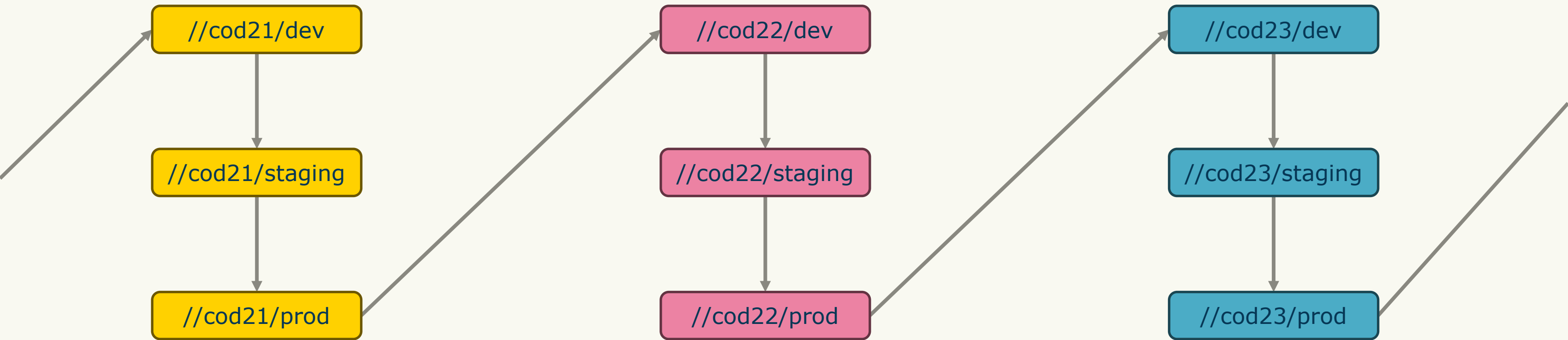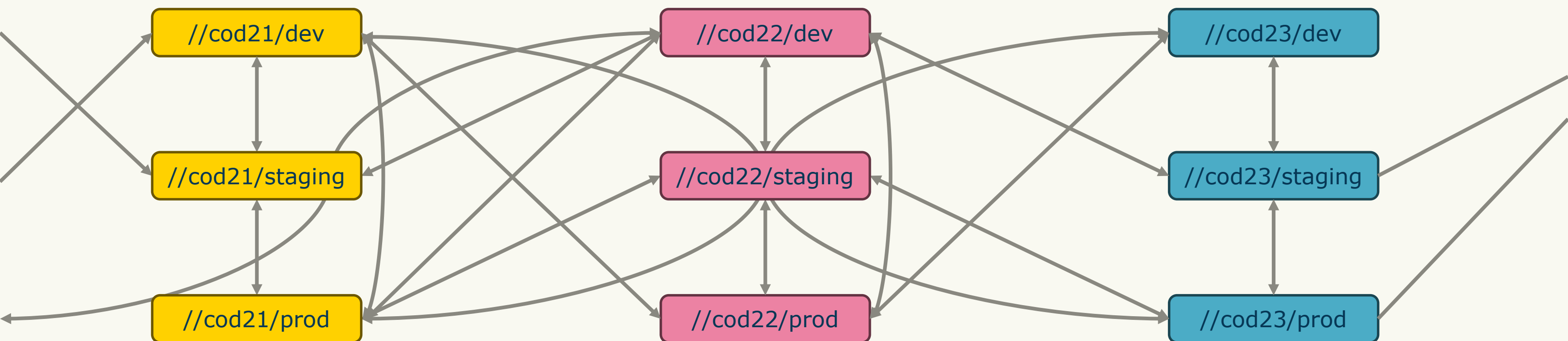# Who am I?

Compass   MergeBot   SENTRY

Agenda

- Problem Statement
- Prior Approaches
- Desired Features
- Mergebot Overview
- Lessons Learned
- Future Plans
- Questions
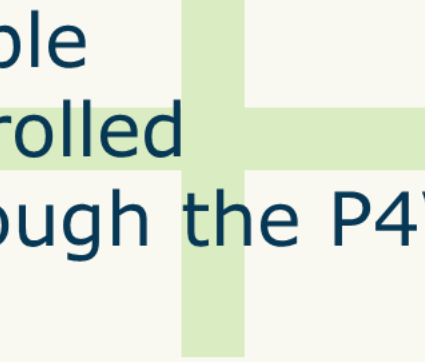
Our (ideal) integration needs

Our (real) integration needs

Previous Approaches

Local Integrations
• Easily customizable • 100% User Controlled • Can be done through the P4V UI
• Need constant user communication • Resolves must be done by one user • Can be memory and disk intensive • High latency

Previous Approaches

Custom CI Jobs

- Easily monitored by everyone
- Can be scheduled
- Virtually unlimited resources

- Resolves require access to the integration worker
- Need different scripts per integration
- No access to dedicated UI
- Not resumable

Desired Features

- Universal UI accessible by everyone
- Easily customizable integrations
- Conflict resolution by multiple users at the same time
- Efficient logging
- Changelist cherrypicking
- Testing the changes before submitting them in the CI
- History records of past integrations
- Integration triggers based on user input, cronjobs and commit tracking

Requirements and Features

Configuration

- Yaml files with strict schemas
- Auxiliary toggles in the UI for specific features

Conflict Resolution

- Installable app (Resolvy) that communicates with the conflicts pod
- Email and slack notifications to users responsible for conflicts

Sample Configuration

```yaml
- source: "//cod22/main"
  target: "//cod23/dev"
  view: ["//cod22/main/... //cod23/dev/...",
         "-//cod22/main/audio/... //cod23/dev/audio/..."]
  reviewers: [test_user_8]
  integration_mode: integrate_and_commit
  impersonate_user: True
  presubmit_url: https://compass.activision.com/api/presubmit/
  lock_stream: False
  resolver: mergebot.resolve_plugins.to_cod23.resolve
  cron: '0 0 * * *'
```

Mergebot UI (Overview)

Mergebot UI (Detailed View)

Mergebot UI (Custom Integration)

Requirements and Features

Architecture

- Backend: Python w/ Flask
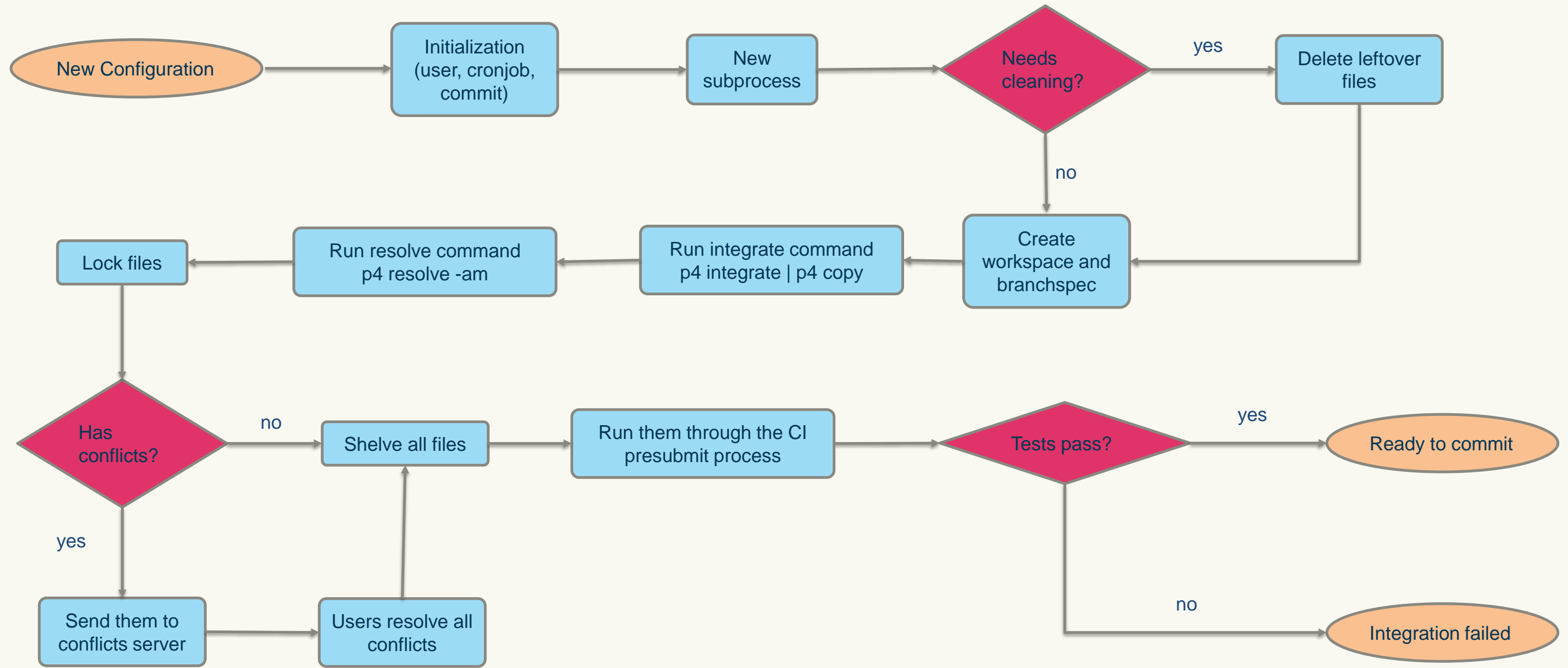- Frontend: JS w/ ReactJS (One Page App)

Metrics / Telemetry

- Sentry for crash reporting
- Prometheus/Grafana for resource monitoring
- Elasticsearch/Kibana for logging

Requirements and Features

Kubernetes Orchestration

- Main pod for running the integrations.
  >100GB RAM, >32 Cores CPU, ~10TB Disk

- Redis pod for integration historic metadata + user info

- Conflicts api server

- Mysql pod for conflict data

- Elasticsearch for storing logs

Sample Workflow

New Configuration → Initialization (user, cronjob, commit) → New subprocess → Needs cleaning?

Needs cleaning? — yes → Delete leftover files

Needs cleaning? — no → Create workspace and branchspec

Delete leftover files → Create workspace and branchspec

Create workspace and branchspec → Run integrate command p4 integrate | p4 copy → Run resolve command p4 resolve -am → Lock files

Lock files → Has conflicts?

Has conflicts? — no → Shelve all files

Has conflicts? — yes → Send them to conflicts server → Users resolve all conflicts → Shelve all files

Shelve all files → Run them through the CI presubmit process → Tests pass?

Tests pass? — yes → Ready to commit

Tests pass? — no → Integration failed

Conflict Resolution

- Arguably the most important functionality

- It is applied in 4 parts:

1. p4 resolve –am

2. custom user script resolution

3. dedicated tools

4. local user resolves

- Optimizing methods 2 and 3 minimizes the time needed for 4

Resolvy

Resolvy (cont'd)

Resolvy (cont'd)

Presubmit Testing on Compass

- Testing is an integral part of Call of Duty
- In-house CI platform (Compass) needs to verify changes
- Not every integration should be committed after resolving conflicts
- Not every integration can pass the presubmit tests
- 2 solutions:
1. Revert the integration and fix the issues on source branch
2. Commit the changes and start work on fixing the issues on target branch

Lessons Learned

- Perforce monitoring is crucial
- Overconfiguration can become an issue
- Not everything can be automated
- Not everything can be predicted
- Fine-tuned alerts WILL save time and money
- Reliability always beats speed

Current State of Mergebot

279 Different Configurations

112 Active Configurations

>30 Configuration changes per week

>400 Successful Merges per week

>10TB worth of files integrated per week

Future Plans

- Improve custom integrations

- UI revamp

- Containerize individual integration jobs

- Stale configuration archiving

Special Thanks

Jan Van Valburg

Bryan Nogradi

John Goldsworthy

Daniel Gleason

The Central Tech team

Questions?