

mojaloop

Versioning

P12 - October 2020

Lewis Daly,
Matt de Haast,
Matt Kingston,
Samuel Kummary

mojaloop

Overview

1. Objectives Recap
2. Progress Update
 - a. Mojaloop Version
 - b. Release Plan + Support
 - c. Branching + Security Updates
 - d. ml-operator
3. Demo
4. Roadmap for PI 12

PI-11 Objectives

- 1. Release a functional v1 of Mojaloop Operator**
- 2. Migrate Mojaloop codebases to semantic versioning**
- 3. Evangelize and Harmonize the Mojaloop Version**

PI-11 Objectives

1 **Complete** Release a functional v1 of Mojaloop Operator

2 **Complete** Migrate Mojaloop codebases to semantic versioning

3 **Complete** Evangelize and Harmonize the Mojaloop Version

Progress Update

Semantic Versioning

Migration of Mojaloop codebases to semantic versioning

1. Adopted across the community
2. Adopted by all services, repos including Helm RC
3. Starting to see more repos adopt automated versioning based on commit messages
4. *Perhaps this is something to adopt across all Mojaloop Codebases?*

Mojaloop Version

Evangelize and Harmonize the Mojaloop Version

1. Current Mojaloop version **v1.1**
2. Focus this PI has been on moving towards SemVer
3. Transitioned [Admin API](#) from x.y.z model to **x.y** model to confirm to the new standard (currently 1.0)
4. Transitioned [Settlement API](#) from x.y.z model to **x.y** model to confirm to the new standard (currently 2.0)
5. Version numbered branching strategy, to support maintenance

Release Plan + Support

Support for Mojaloop (Helm) Releases

1. No formal policy currently, most recent release being supported;
Options - ML Foundation, competitive space
2. Ensure support for previous releases to encourage adoption and support existing implementations / deployments
3. Standards guidance is in place regarding general release mechanisms, hotfixes, bug logging & triaging but, policy needed on support

Branching + Security Updates

- Most of our work today is done on *master*
- With this strategy, master will only ever contain the *latest* changes
- We need a plan for applying security patches and bugfixes to multiple supported versions
 - **security patches:** dependencies, docker image updates
 - **bugfixes:** something broken in our code

Our Branching Strategy

- Based heavily on [releaseflow](#)
- Normal everyday releases are made from *master* with the format [vx.y.0](#)
- We maintain a list of “supported versions”
- For a given supported version, we create a release branch with the format: [release/vx.y.z](#)
- patches and bugfixes are applied to that release branch, and a new version is released, incrementing the z version

Proposed Branching Strategy



Supported Versions:

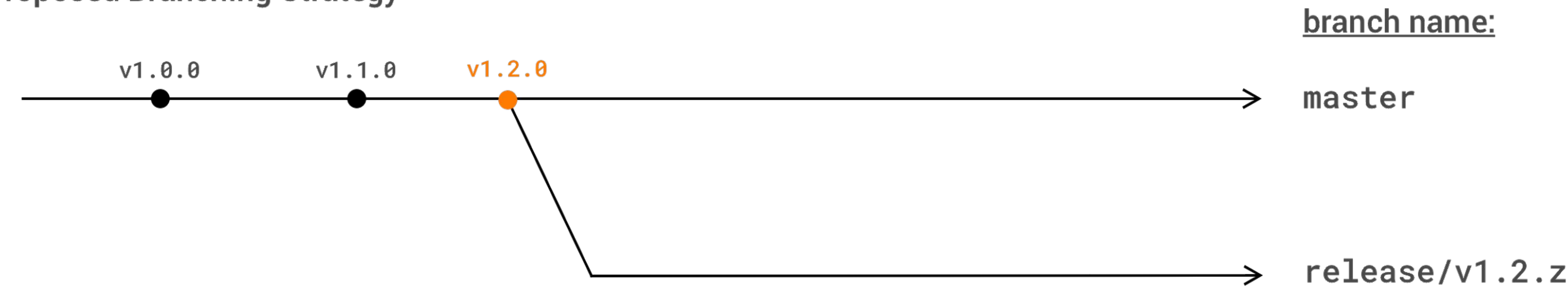


Proposed Branching Strategy



Supported Versions:

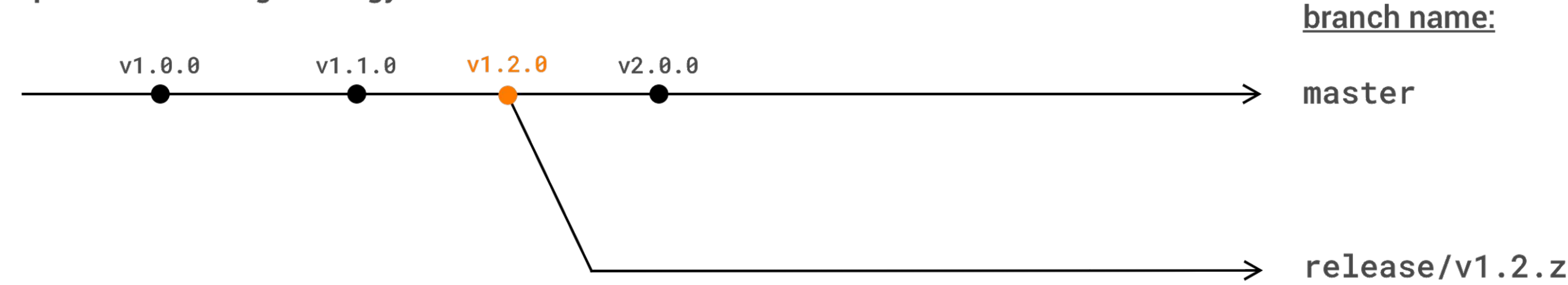
Proposed Branching Strategy



Supported Versions:

- v1.2

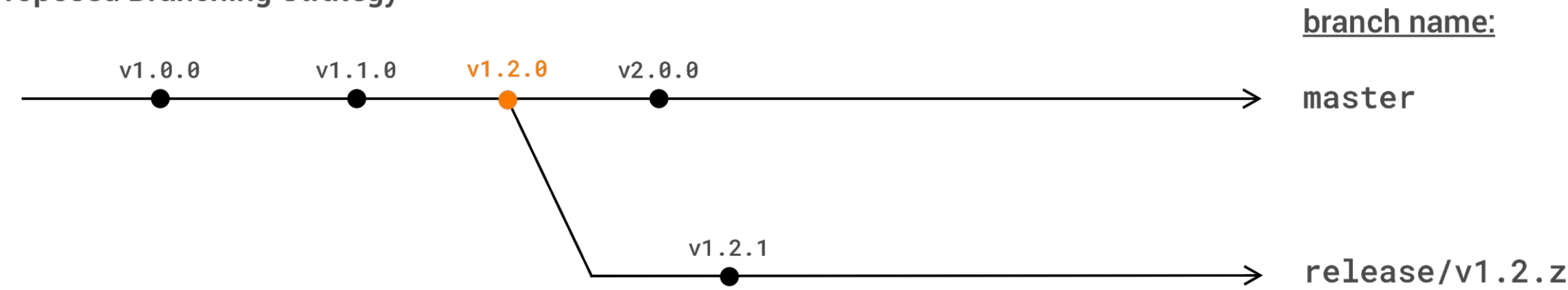
Proposed Branching Strategy



Supported Versions:

- v1.2

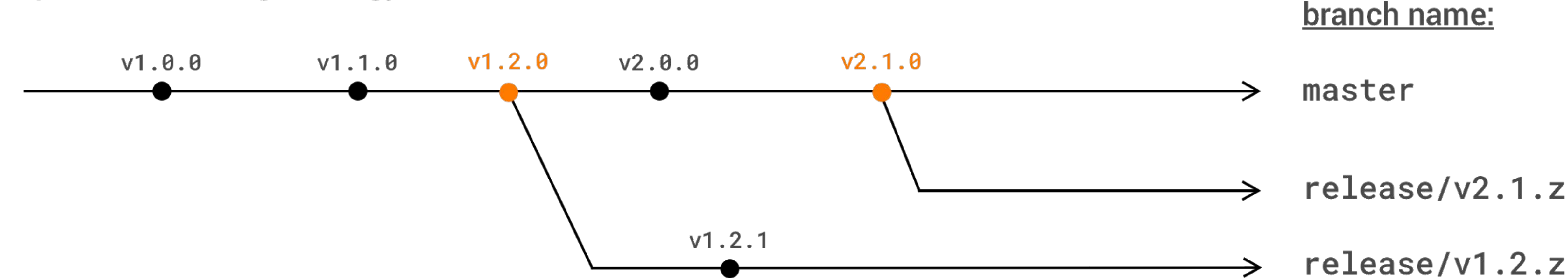
Proposed Branching Strategy



Supported Versions:

- v1.2

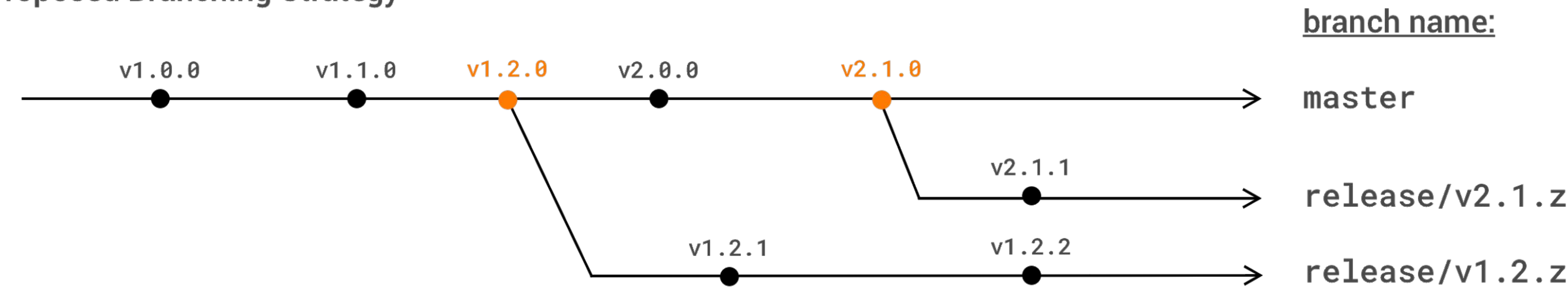
Proposed Branching Strategy



Supported Versions:

- v1.2
- v2.1

Proposed Branching Strategy



Supported Versions:

- v1.2
- v2.1

ml-operator v1.0

ml-operator

- An *operator* is a [Kubernetes Design Pattern](#)
 - A custom extension to Kubernetes
 - Allows us to insert business logic into the running of our environment
 - e.g. “Perform backups of application state on a regular basis”, or “Check that the image versions of my pods are the latest”
- ml-operator v1.0:
 - regularly scan mojaloop images for updates based on an upgrade strategy
 - alert operators of available upgrades
 - provide instructions for how to apply those updates

image-watcher

- Companion to the ml-operator
- A simple api for watching published images in a docker registry (in our case, docker-hub)
- Takes an image name, image version, and upgrade strategy
- Gives you the latest version.

```
GET /image/mojaloop/central-ledger/v11.0.0?strategy=bugfix
Content-Type: application/json

{
  "tag": "v11.0.1",
  "fullImage": "mojaloop/central-ledger/v11.0.1"
}
```


ml-operator



incoming-webhook APP 4:16 PM

ml-operator - all monitored services are up to date



incoming-webhook APP 4:30 PM

ml-operator - The following services are not on the latest secure version and should be updated:

- `mojaloop/account-lookup-service`: latest version is `v11.1.2`
- `mojaloop/auth-service`: latest version is `v11.2.7`

Use the following commands to patch your deployments with `kubectl`:

```
kubectl patch deployment account-lookup-service --patch '{"spec": {"template": {"spec": {"containers": [{"name": "account-lookup-service", "image": "mojaloop/account-lookup-service:v11.1.2"}]}}}}'
```

```
kubectl patch deployment auth-service --patch '{"spec": {"template": {"spec": {"containers": [{"name": "auth-service", "image": "mojaloop/auth-service:v11.2.7"}]}}}}'
```

Demo

PI 12 Objectives

PI 12 Objectives

- Implement auto-releasing for our code pipelines
 - Improve release notes
 - Consolidate CI/CD Config across all repos
- ml-operator v1.1
 - Apply image patches upon confirmation
 - Nice UI for hub operators
 - Automatically apply upgrades during an upgrade window
- Data migrations + Upgrades
 - Revisit our zero downtime deployment POC