GitHub Applets

Brainstorming session on enabling an applet-inspired Blueprint approach based on Fission Workflows

Recap: what's Fission again?

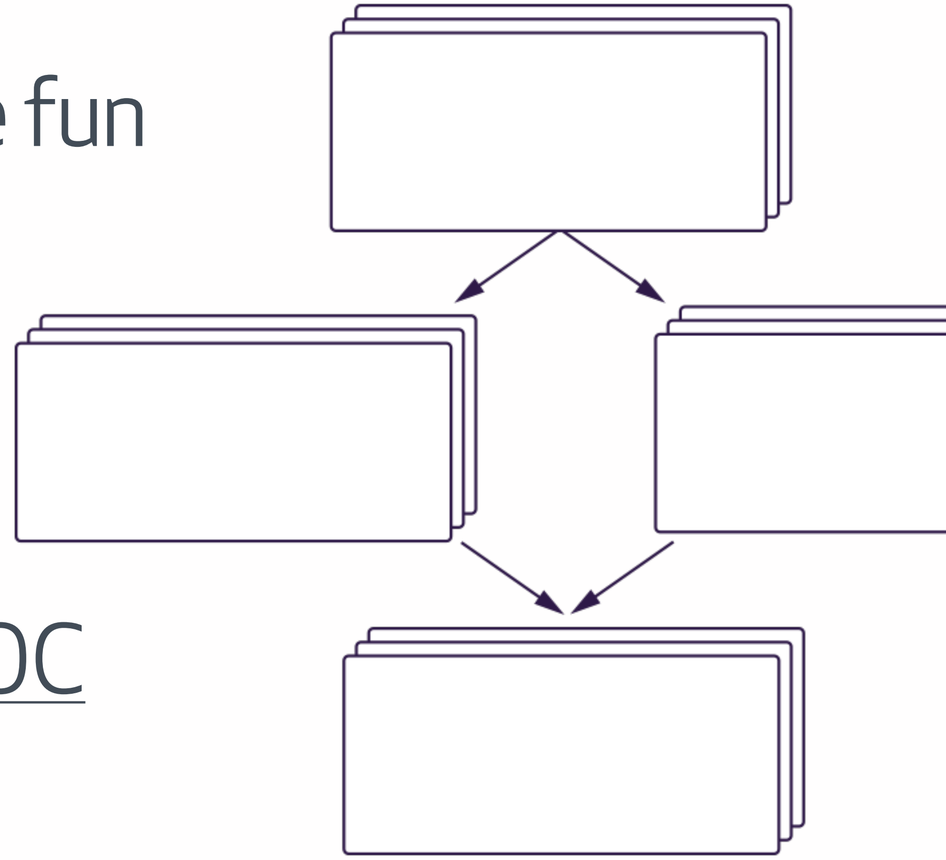
- **Fission** is a framework for serverless functions on K8S
- Functions are code that gets triggered by HTTP requests OR events
- Functions are deployed **instantly**
- No container maintenance needed!
- ✨An awesome find by Lee ✨

Fission Workflows

- Workflows are a sequence of inter-dependent **functions**
- **Triggered** like normal functions by a Controller
- Backed by NATS Messaging system (pub-sub, persistent)
- Orchestrated by a (stateless) Scheduler
  - State is re-constructed from **events** (reactive)

Benefits of Fission Workflows

- 🔓Enable custom, non-linear workflows on GHE!
- 🎢Safe: we create guardrails, you have fun
- 🛳Ideal for rapid iterations
- 💵Virtually zero maintenance
- ✔As already demonstrated by Lee's POC



What about Probot?

RefImpl for GitHub Apps

- NodeJS only
- No workflow model
- Deployment is manual with no HA/DR
- Stateful services are hard to implement
- Express wrapper to Function which is Fission OOTB

OK, so...?

Let’s take this a step further

- What if…anyone could create **integrations**?
- What if…anyone could easily **create** workflows?
- What if…anyone could **publish** their workflows?
- What if…partners could **assemble** and **reuse** blueprints?

What if we took an “applet” approach?

- Powered by Fission workflows
- With support for custom integrations
- Subscribe to events
- Can define payload templates
- Define internal logic to handle complex point-to-point comms

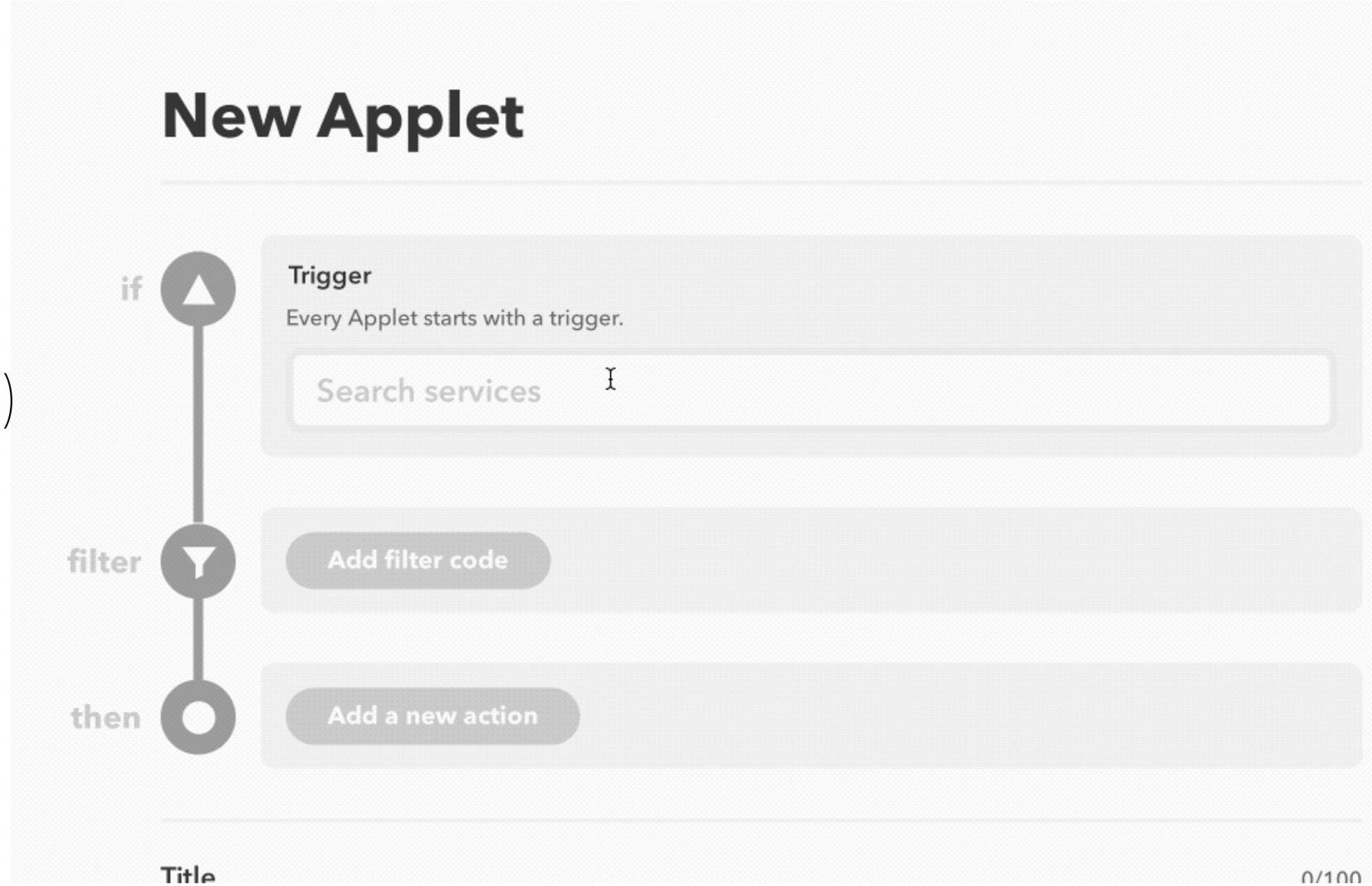
Let's visualise all of that.

The case for GitHub Applets

> "As a **devops eng** working for one of GitHub **partners,**
>
> I would like to **easily** create applets
>
> To deliver **highly specialised** solutions
>
> to our customers **effortlessly**"

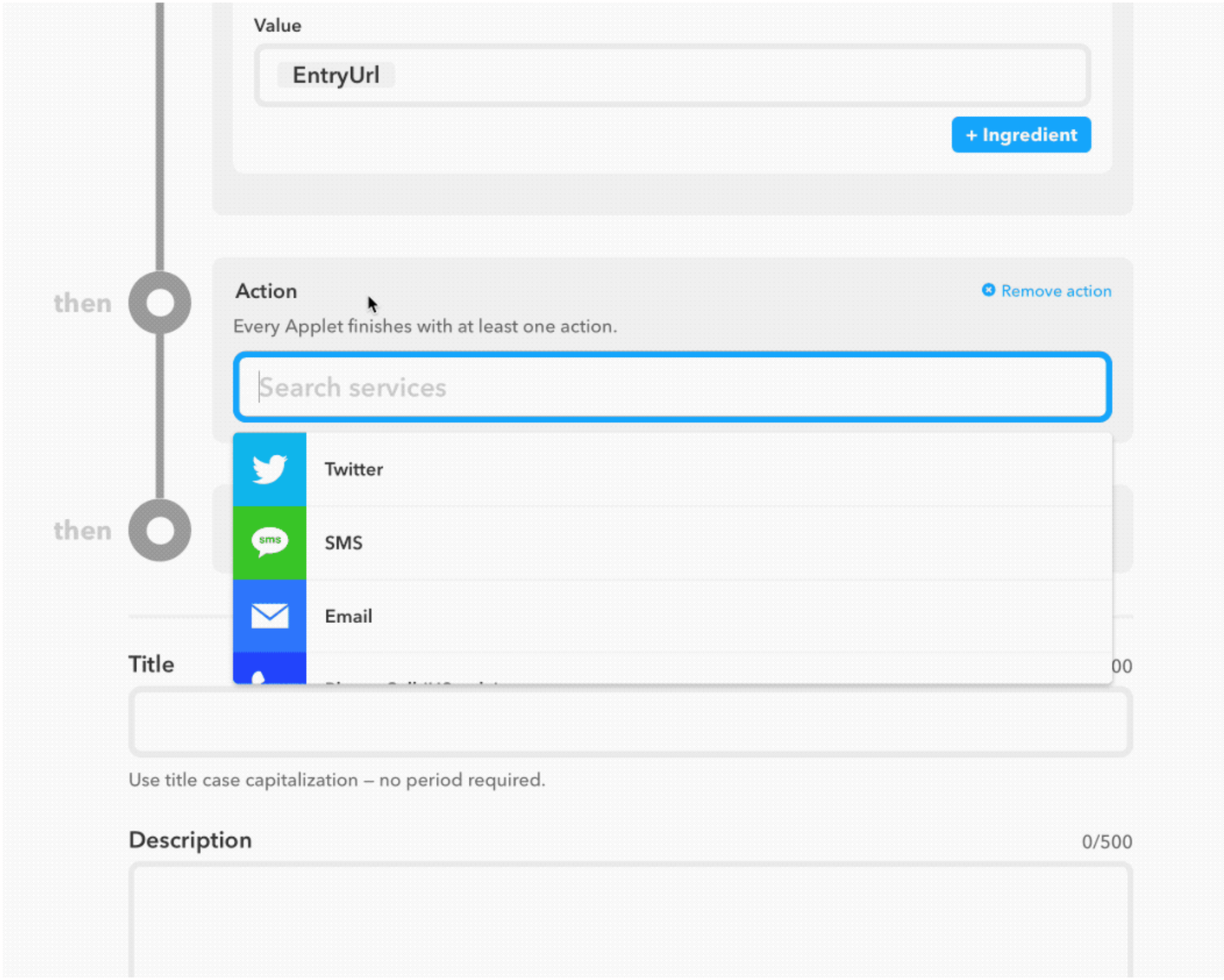
1. Spawned by **triggers**
2. Triggers = **Events**

   (webhooks + Fission Events)
3. Defined via **integrations**
4. **Custom** Payload

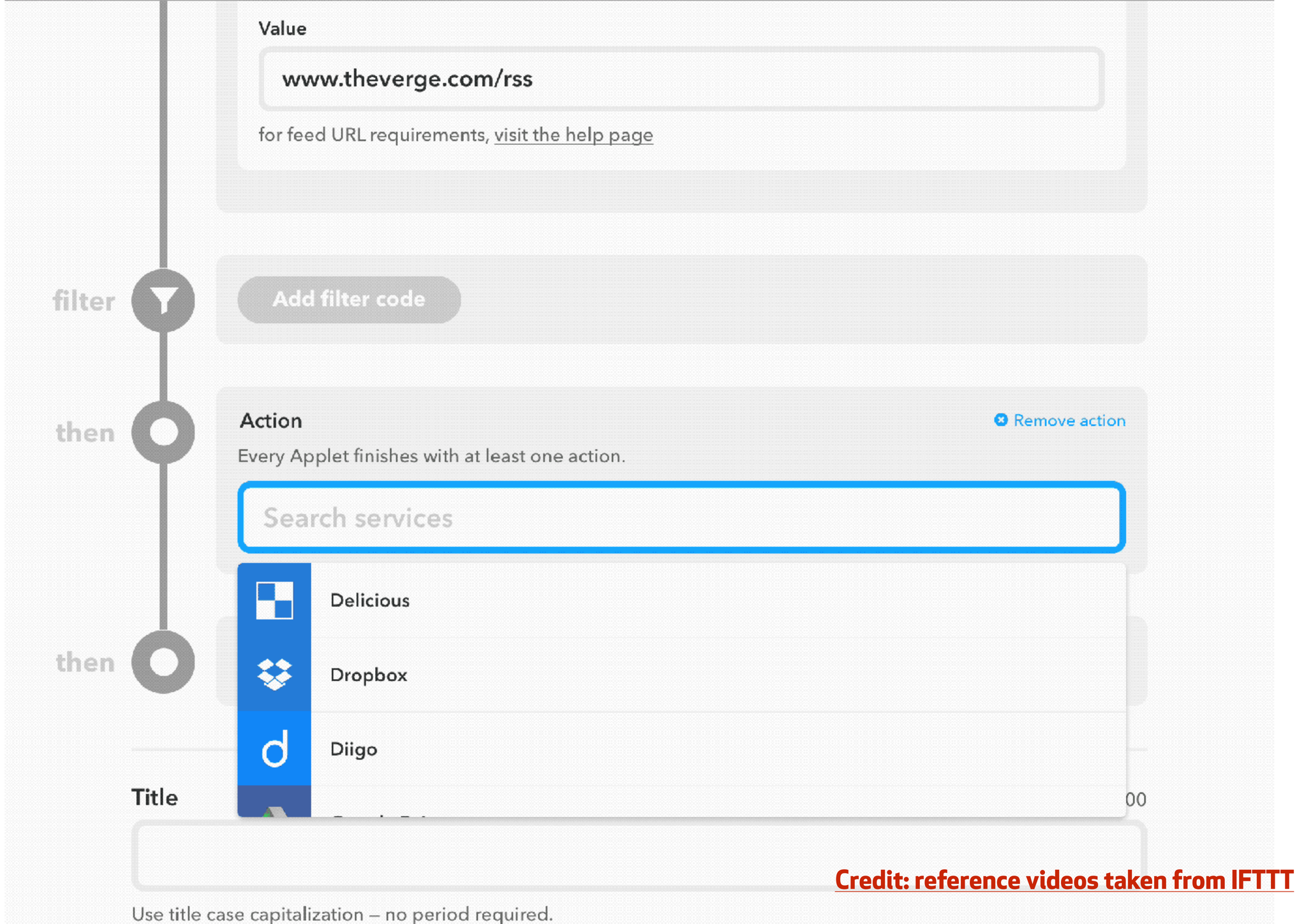


Credit: reference videos taken from IFTTT

1. Actions defined by **integrations**
2. Custom integrations (e.g. a store)
3. Official GitHub action packs
4. Reusable & **composable**
5. Configurable



Credit: reference videos taken from IFTTT



Credit: reference videos taken from IFTTT

Replacement for GitHub Services?

https://developer.github.com/changes/2018-04-25-github-services-deprecation/

How cool would that be?

Imagine the possibilities...

🎰🎲📺 Time for a random “Crazy Eights” round! 🎰🎲📺

When a PR is merged and the target branch is “production” run SNYK on the HEAD of the branch and store the report on Auditology.

by Kiratech S.p.A

Security Auditing ...

Map your JIRA workflows to GitHub and collect Team metrics on an Organisation-wide dashboard on Amplitude

by Amazic GbV

Security Alerting ...

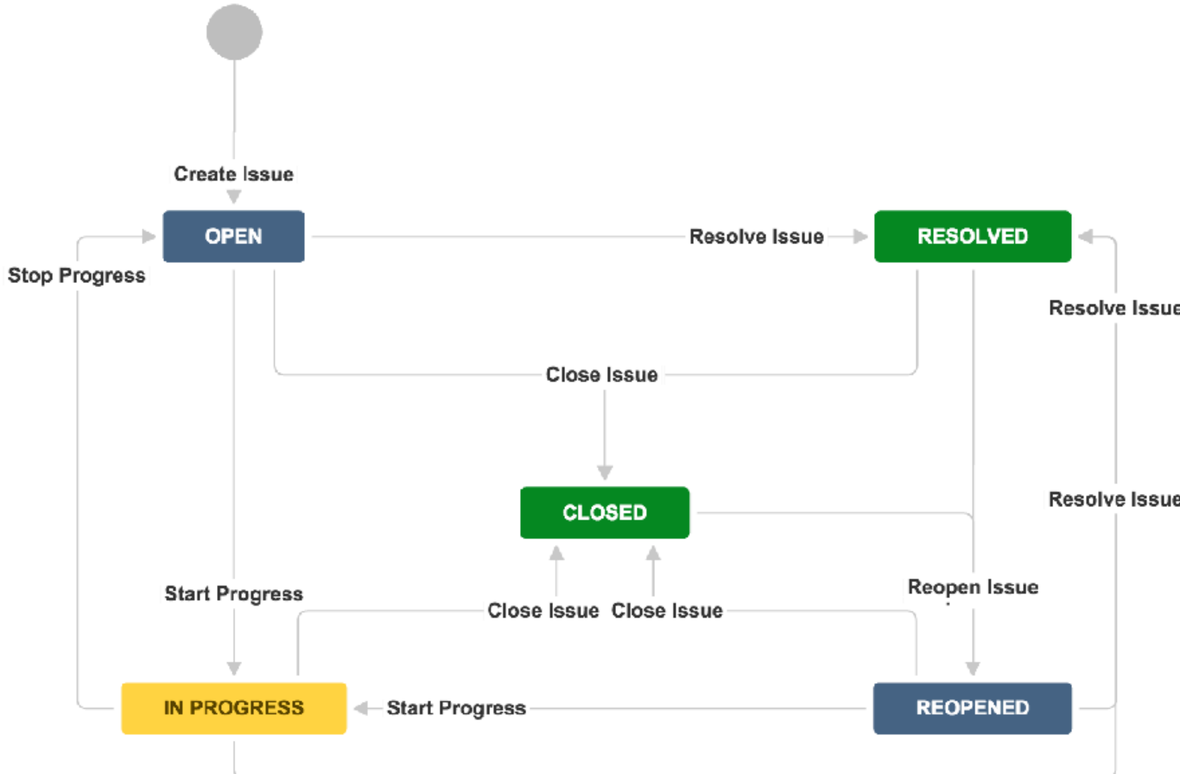
When a merge to “master” occurs, run XL Release to deploy to staging and call the GH Deployment API, otherwise deploy with XL Deploy to Test and run cypress.io against the changes and call the checks API where there are errors

by XebiaLabs

Ideas Brainstorming

Demo

Possible Next Steps

Things we should POC

- Do It Yourself Continuous Integration
- JIRA Broker for Enterprise
- Probot OOO and Stale
- Others…