INCUBATING PROJECTS

THE BIG QUESTIONS…

- We have identified a few projects, but how do we actually **get started** with them?
- When do we **stop** following something which looked like a good idea but it's not, possibly avoiding blame?
- How do we ensure we have something to test **sooner rather than later**?
- How do we share the expertise / resources to create a community service, rather than a "tool" someone else has to deploy?
- How do we remain flexible and decentralised, allowing different needs to coexists?

WHAT DO I NEED AS A TOOL DEVELOPER TO START A PROJECT?

- For small / medium scale projects, source-code management, communication and CI are currently well served by "**freemium**" products (Github, Travis, Circle-CI, Slack, Pypi).
- A viable infrastructure solution for doing deployments (think **Heroku**-like, or **kubernetes**-like) to some hundreds of users, in a lab independent way, is not really commonly available. Freemium solutions are not such when you start building a service.
- I need an easy way to **publish to CVMFS.**
- I need expertise on how to **provide a service** to the community (i.e. cross lab authentication, user management, multiuser design suggestion), not a standalone bit of software.
- I probably need **good and sincere feedback** on whether what I am doing is useful for the community.
- **Recognition of my work as valuable for my career.**

WHAT AM I WILLING TO GIVE AWAY AS DEVELOPER?

- Soul (e.g. "Powered by HSF" :-)).
- **License**.
- Some control on design decisions on areas related to the community (e.g. "use this / that authentication strategy", "use pip to distribute python modules").
- Instructions for **automated builds / deployments**.
- **RFC** on future features of a given product.
- **Progress reports.** Usage metrics.
- Understanding that if a project is not used it loses access to "incubator resources".
- Understanding that if a compatible freemium / OpenSource solution arises, we should switch to it for the sake of the community.