



Discovering and balancing  
multi-cloud applications

**<http://innotech.github.io/hydra/>**





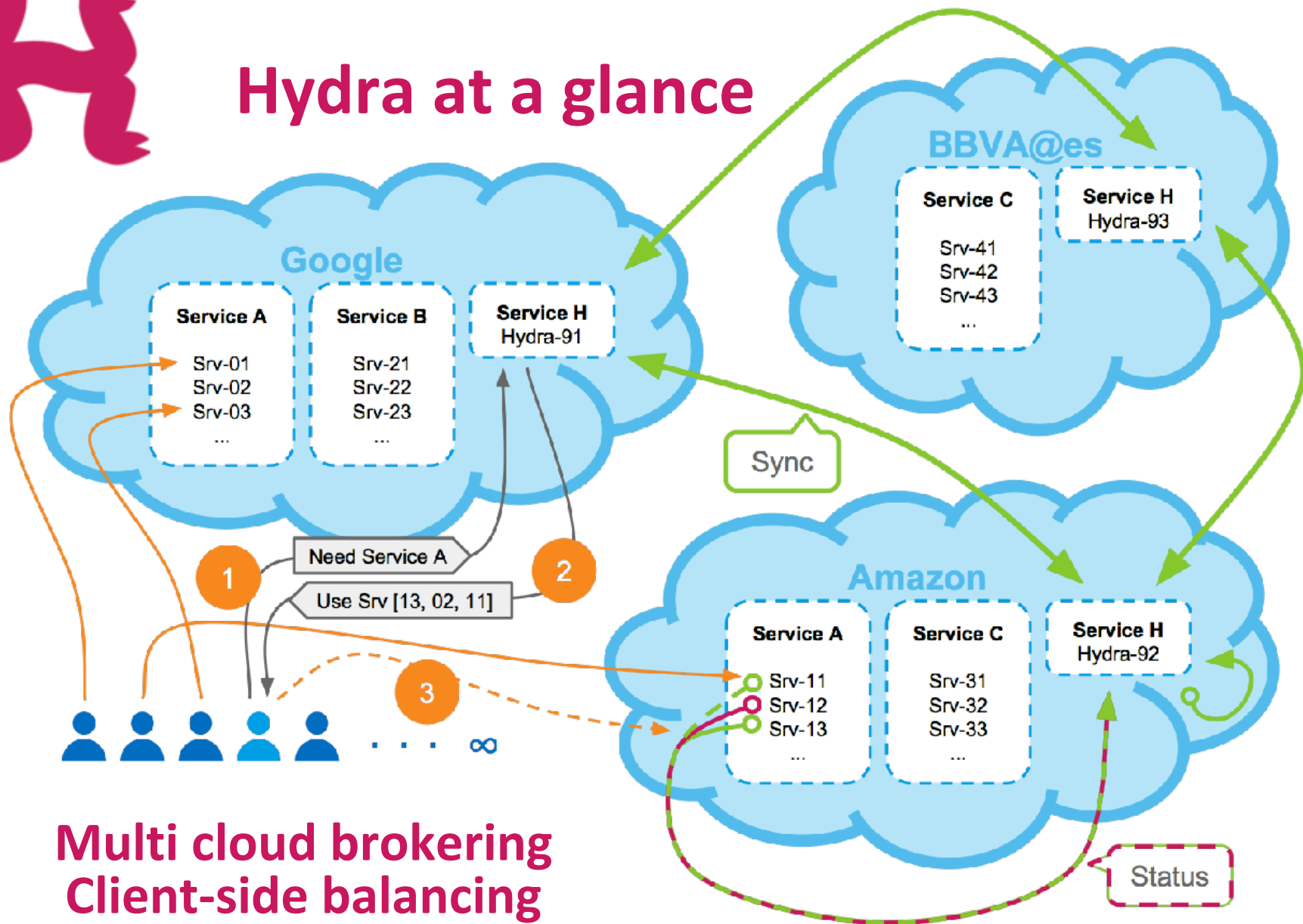
## Why Hydra?

- We have several cloud providers options and we do not want to marry with any of them
- We want to build web-scale applications able to give world wide service.
- We want our application be boom resilient.





# Hydra at a glance



**Multi cloud brokering**  
**Client-side balancing**





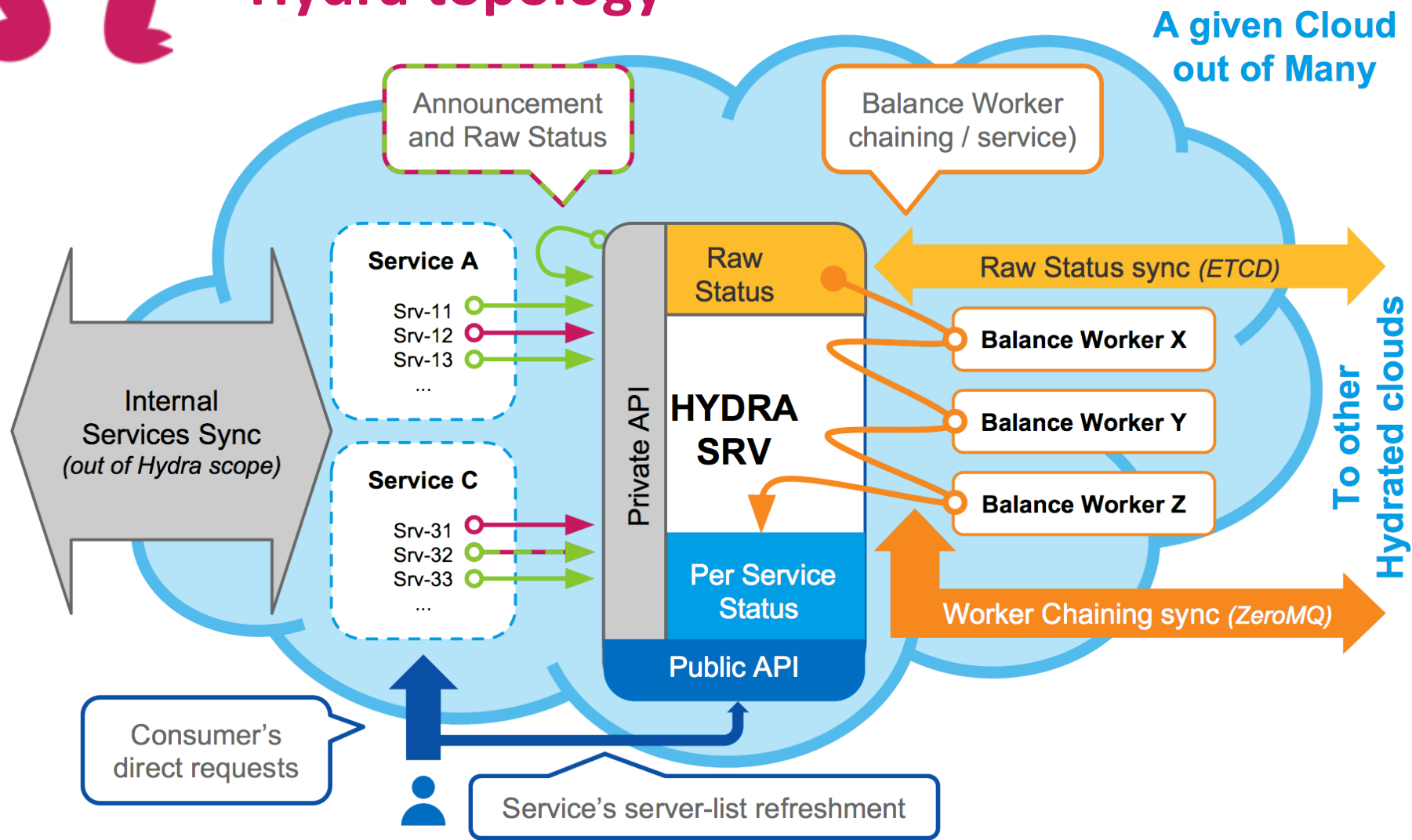
## How does Hydra works?

- Collects data by monitoring each individual service server.
- The data is processed and stored in the Hydra server.
- Every Hydra server share information with other Hydra servers.
- Clients ask for address information about a service to an Hydra server.
- Clients directly connect with the right service server.





# Hydra topology





# Live demo!!

<http://innotech.github.io/hydra-web-monitor/>





## Advanced use case: Zero Downtime

- Deploy new servers
- Configure Hydra to give address information in order to clients go to new servers
- Wait for old server to be idle
- Shutdown old servers





## Advanced use case: A-B Testing

- We have some servers with A version of a service
- Deploy new servers with B version of the same service
- Configure Hydra to give address information of B servers to a **controlled group of clients**
- Hydra will continue giving address information of A servers to the rest of the clients
- The controlled group of clients test the B version of the service







# Conclusions

- Unique capability of Multi-Cloud balancing.
- Client-side balancing without expensive hardware or premium additional services.
- Extremely flexible from all points of view
- Only one hydra server per cloud is enough.





## Future work

- User management, multi-tenant security layer.
- Improve system monitor.
- Improve hydra server stability.
- More Hydra workers.
- Performance tests.





# Any Question?

<http://innotech.github.io/hydra/>

