SpringOne Platform by Pivotal

Feature Toggling With FF4J

By Sasi Peri

October 7–10, 2019
Austin Convention Center

Agenda

1. Why I am doing this presentation?
2. Use cases
3. Demo
4. Pitfalls
5. Reference links, source code (GIT)
6. Q & A

Safe Harbor Statement

The following is intended to outline the general direction of my findings and is intended for information purposes only. You are encouraged to do your own thorough analysis and research before the use of the technology in your production grade applications.

Definitions

- **What's a feature?**
  - A user story that's visible OR a requirement resulting in backend code(logic).
  - Also known as "feature flip", "feature flags" or "feature bits"
- **What's feature toggling?**
  - Feature toggling is a technique used to enable or disable certain behavior of the system (typically at runtime) to gradually release and test new features.
- **What's FF4J?** *(**Cedrick Lunven**)*
  - FF4J stands for "**f**eature **f**lipping for **j**ava". This open source framework in Java (licensed under Apache-V2), works very well with any Java and(or) Spring/SpringBoot based applications.

Note: *Other frameworks in java/other-technologies are in the reference section*

Technology

- Java, open source
- Apache-v2 license
- Spring, SpringBoot, Non-Web, DeskTop and any such applications
- REST-API available for non-java.
- AOP based and Non-AOP (IOC) toggles available.
- JSP Tag libs and ThymeLeaf Dialects available.
- Can integrate with rules engines like **Drools**

Uses (cont'd...)

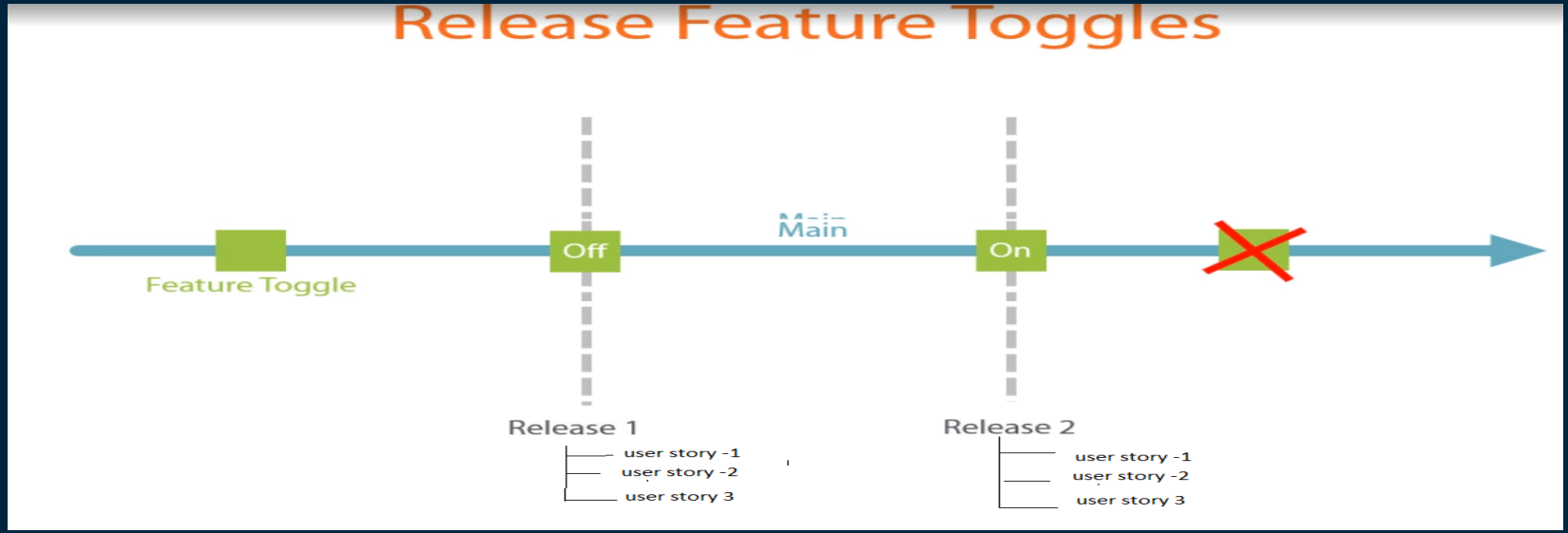
- Release and deploy based feature toggles (short lived). Examples include
  - Hide feature that are not "complete", business changed the mind last minute, bug identified right after deploy in a feature.
  - Canary releases
  - Blue-green deployments
  - A/B testing
- Business feature toggles (usually long lived). Examples include
  - Dark Launch
  - Business functionality flip on the fly
  - Business rule-based feature enabling
  - Many other strategies to gradually release functionality into production (Role/Group, Region, Clients/Servers) .

Flipping strategies

- Simple flip & group (release) flip
- Role based flip
- Client & Server Filters & Region based
  - Blacklist
  - Whitelist
- Release date based
- Office hour strategy
  - Week timetable
  - Public holidays
  - Special openings
- Ponderation & Dark Launch Strategy *(% of requests based, Dark Launch zero downtime deployment pattern)*
- AOP-Based flipping *(that can be integrated with Circuit Breaker)*
- Custom Strategies

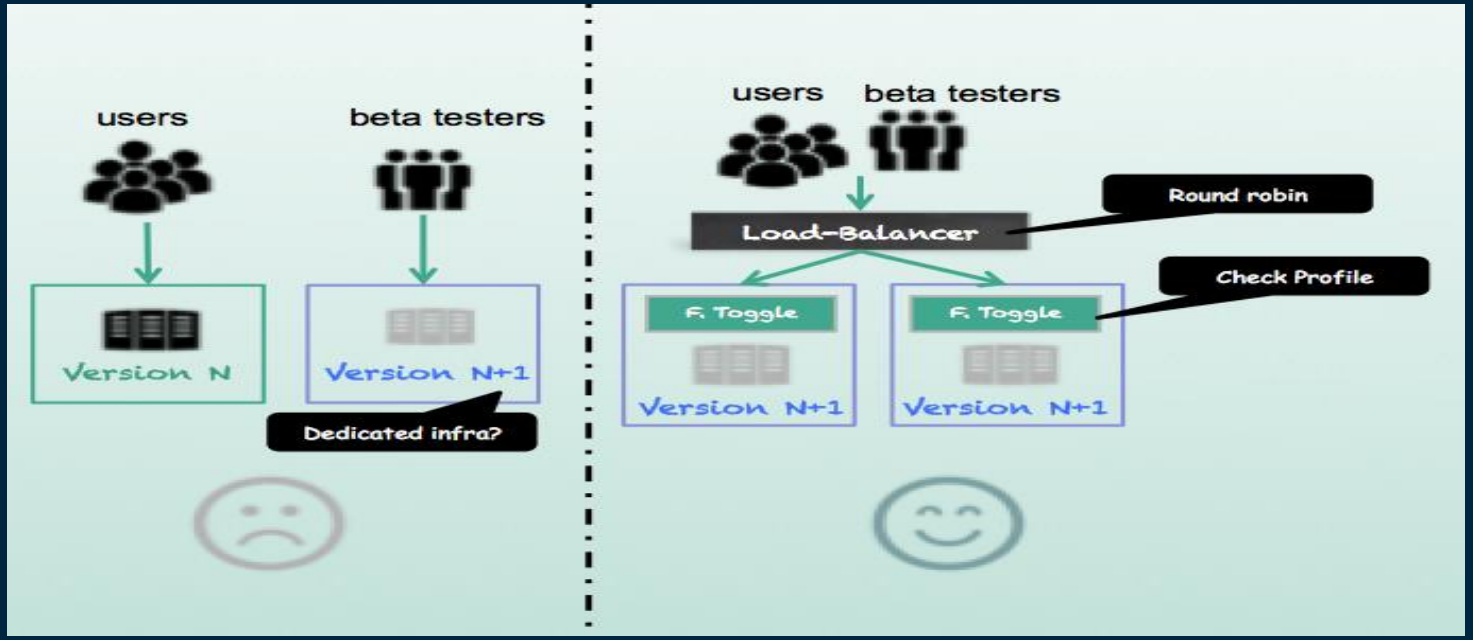
Uses

- Release and deploy based feature toggles (short lived)
  - Hide feature that are not “complete”.
  - CI & CD made easier
  - Reduce the cost of long-lived branches
  - Avoid merging hassle
  - Short lived (must be removed as soon as release is ready)



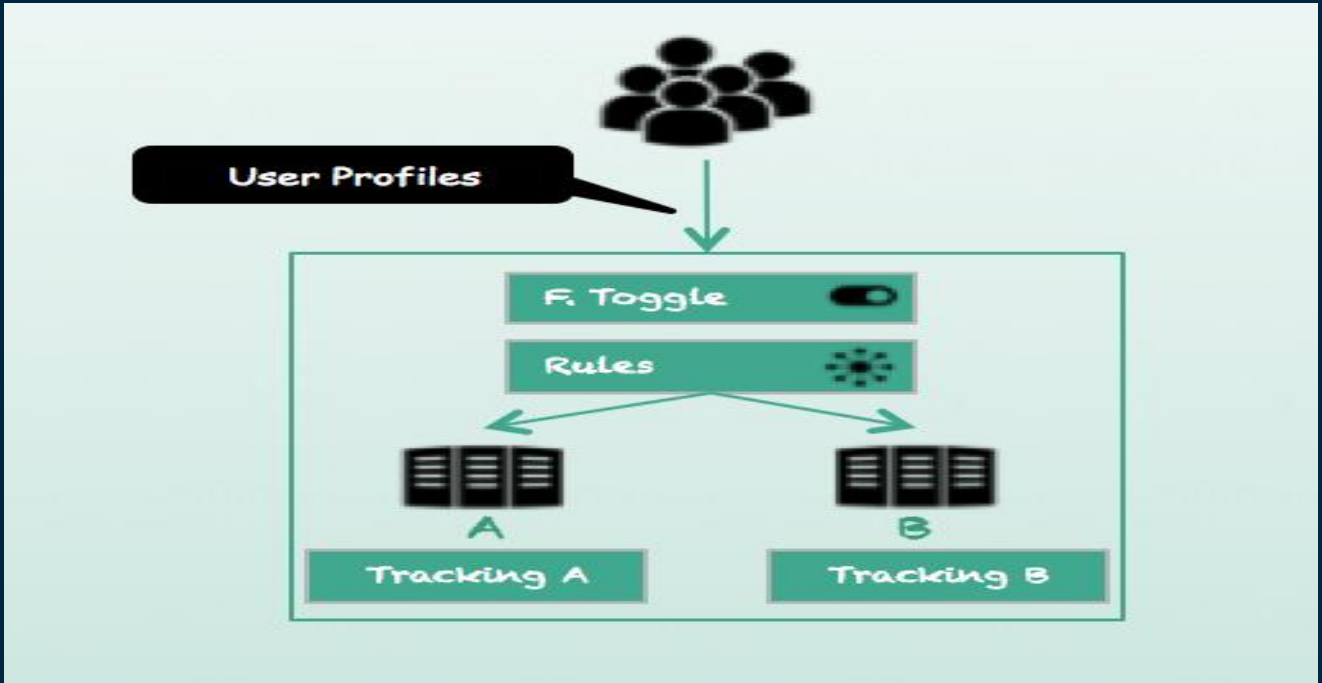
Canary release

Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.



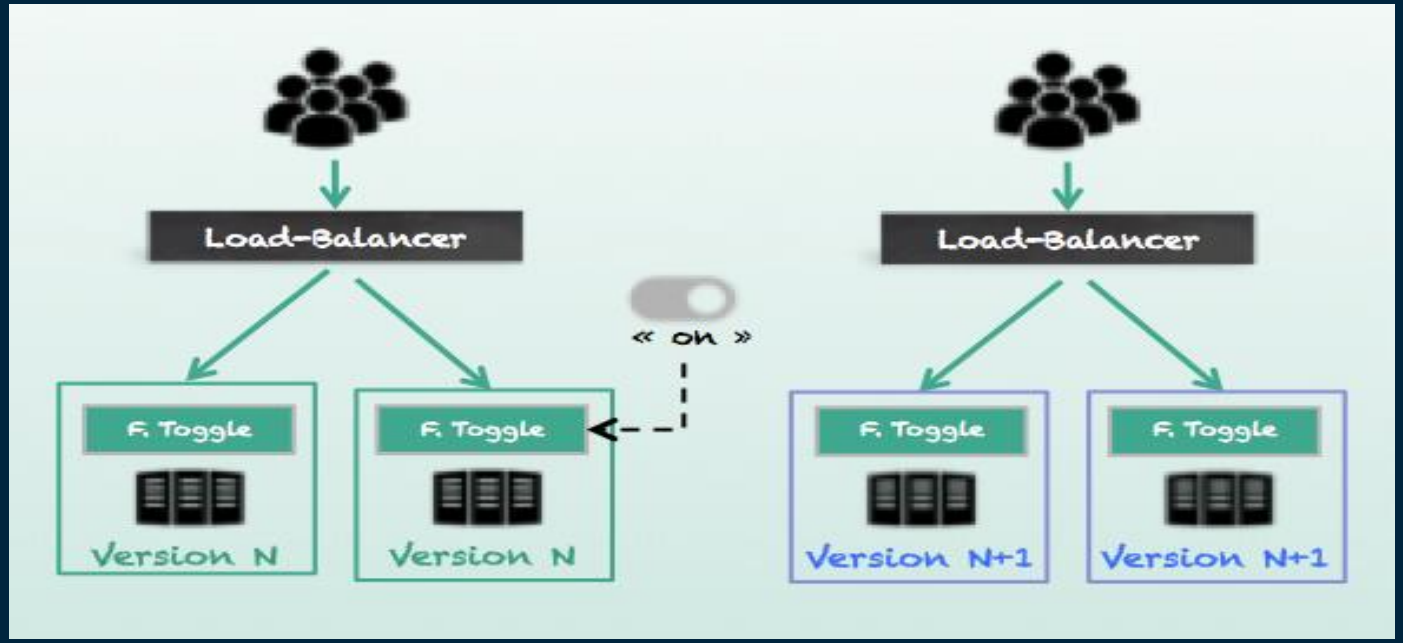
A/B test

A/B testing (also known as split **testing** or bucket **testing**) is **a** method of comparing two versions of **a** webpage or app against each other to determine which one performs better.



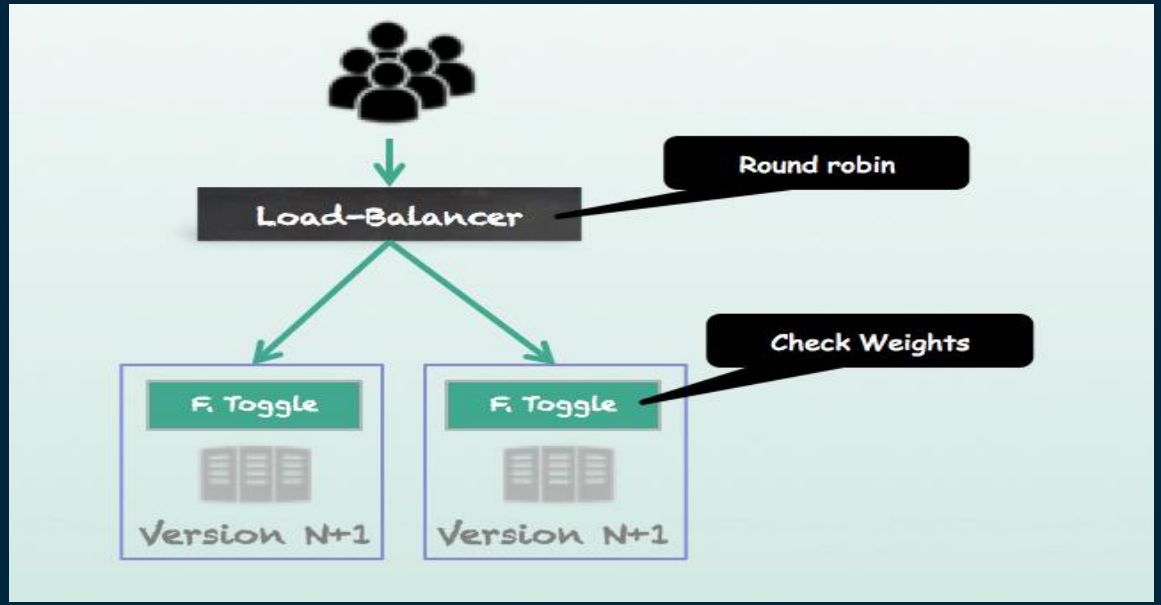
Blue-green deploy

Blue-green deployment is a technique that reduces downtime and risk by running two identical production environments called **Blue** and **Green**. At any time, only one of the environments is live, with the live environment serving all production traffic. For this example, **Blue** is currently live and **Green** is idle.



Dark Launch

The Dark Launch develop pattern is the capacity for a system to enable a new feature for a subset of incoming requests and measure if the new feature introduce some overhead. Without feature toggle you must deploy your new package on a node of the cluster and measure. With this strategy all nodes of the cluster have the new version and execute new behavior for a subset of requests.



DEMO
(SpringBoot & FF4J)

Feature and property stores

Feature

- Simple XML based
- Caching
  - JCahce
  - Redis
  - EHCache
  - ...and more
- RDBMS (supports any DB via JDBC)
- NoSQL
  - Cassandra
  - Mongo
  - Elastic search as NoSQL
  - Neo4j (Graph DB)

Property

- Spring Config Srv
- Archaius/CommonsConfig
- ..and more

Audit and monitor

Auditing

All CRUD operations on features and properties can be audited.

Monitoring

Able to record all features usage (hits) and compute metrics based on different KPI like users, sources or hosts. Dashboards allow to search and filter in history, asynchronously with no blocking.

Administration

- Embedded web-gui
- Stand alone GUI
- Command Line Interface
- MBeans are provided to operate via JMX.

Pitfalls

*"**Release toggles**" particularly are the last thing you should do, first choice should be to break the feature down so you can safely introduce parts of the feature into the product.*

*- **Martin Fowler***

- **Design & Develop**

  Feature toggle requires

  - A robust engineering process
  - Solid technical design (including test strategy with on/off)
  - A mature toggle life-cycle management

  If not

  - They will become largest and worst technical debt.
- **Testing**

  Need to develop test cases with features ON & OFF.

Thank you

- Jim Shingler - Director, Platform Architecture & DevOps @ Cardinal health
- Ralph Meira – Platform Architect, Pivotal
- Tech Elevator - Cleveland

Links and references

References

1. KNightmare devops cautionary tale(Knight Capital story) https://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/
2. Some feature toggle frameworks list in different technologies http://enterprisedevops.org/feature-toggle-frameworks-list/
3. Feature toggle the worst Tech Debt http://swreflections.blogspot.com/2014/08/feature-toggles-are-one-of-worst-kinds.html
4. https://martinfowler.com/bliki/FeatureToggle.html

Other Links

1. FFJ – Home : https://ff4j.github.io/
2. Source code for this demo(and more free code) is available on my Git Gub: https://github.com/sasiperi/FeatureToggle-Boot2

Q&A