Pivotal

Spring Cloud Dependencies

Spring Cloud

© Copyright 2017 Pivotal Software, Inc. All rights Reserved. Version 1.0

Problem

- Spring Cloud
  - Is an umbrella project
  - It is huge
  - Release Management must synchronize large number of subprojects

Solution - Bill of Materials

- Spring Cloud Introduces the concept of *Release Train*.
- Each Release Train has an associated Bill of Materials (BOM).
- A Release Train (Spring Cloud version) is identified by a name:

```
{Alphabetically increasing name}.SRXX
```

- Example:

```
Greenwich.SR3
```

See http://projects.spring.io/spring-cloud/ for more info

Spring Boot Starters Revisited

- Spring Cloud sub-projects that supply executable applications leverage Spring Boot.
- Use Maven Central or build tool dependency plugin to determine versions

Curating Dependencies

- Spring Cloud BOM and Spring Boot Starters help bootstrap development
- But there are problems lurking:
  - Integrating 3rd parties not already including in Spring project transitive dependencies?
  - Middleware or backing resources on different Spring project versions than client projects?
  - Bloated deployment artifacts (with large number of unused dependencies)
  - Potential masking of viral licenses
  - Potential masking of security vulnerabilities

Curating Dependencies - Solution

- Use Starters and BOMs to get up and running for green-field projects or MVPs
- Before product release (or when introducing 3rd party dependencies outside Spring ecosystem):
    - Analyze project dependencies (direct and transitive)
    - Analyze and resolve dependency conflicts
    - Exclude unnecessary dependencies