Open Source with Open Source: Component Mngmt with SW360

Speakers
Johannes Kristan (Bosch Software Innovations GmbH), Michael C. Jaeger (Siemens AG)
Introduction

“If Software Is Eating The World, Then Open Source Will Chew It Up (And Swallow)”

Adrian Bridgwater
Open Source Licensing

Distributing open source software requires to

• Provide licenses of involved software
• Provide copyright statements of involved authors
• Provide disclaimers
• ... and much more
Open Source Project Info

Code Quality

Vulnerability

Project Liveness

Expertise
What is Needed?

Software Product

License
Liveness
Quality
Expertise
Component
Component
Component
Component

Software Product

License
Liveness
Quality
Expertise
Component
Component
Component
Component

!
Use Cases and Roles

- Ensure Compliance
- Share Knowledge
- Maintain Assets

Architects

Legal Counsels

Sec Officer

Developers and Experts

Quality Managers

R&D Managers

Project Managers

R&D

Managers

Managers

Managers
Central Hub

- License Scanner
- Artefact Repository
- Source Code Scanner
- Code Quality Checker
- Vulnerability Management
- Project BOM Management

SW360

- Collect
- Aggregate
- Correlate
- Enrich

```json
"_embedded": {
  "sw360:components": [
    {
      "type": "component",
      "name": "Angular",
      "componentType": "OSS",
      "createdOn": "2016-12-15",
      "vendorNames": ["Google"]
    }
  ]
}```
Eclipse SW360

- An Eclipse Foundation incubator project
- Eclipse Public License 1.0 (EPL-1.0)
- A portal application
- A catalogue of components, releases
- A catalogue of your projects / products
- A store for attachments of these
- A end user front end to FOSSology
Foundation

Component

Project -> Release

Release -> Vendor

Release -> Licenses
The Data Model is Important

---

Goals and Motivation

○ Reduction of duplicate entries

○ Separating vendor from component names and release tags brings clarity to component naming

○ Interoperation with other systems
  - As such we need to support the CPE standard which also implement this 3-parts separation

○ Having the clear modeling of data enables better search and filtering abilities
How it Works

Create Component Entry
- A component is a container for releases
- Just creating a component alone creates an empty shell

Add Release Entry
- Release = Version
- A component can have multiple releases
- It makes sense to group them
- Now a vendor can be assigned

Add Package (Upload) to Release
- Multiple open source packages of the same releases: multiple releases actually
- There should be just one upload per release
How it Works

- A project is a structure to main a BOM
- Can represent product, service, delivery

- Release = Version of component
  - Projects can contain other projects
  - Allows for building blocks

- Send to analysis
- Upload analysis results
- Generate product level documentation
SW360 Core Features

- Protex™
- FOSSology
- Project Management
- Configuration Management
- Vulnerability Database
- Code Quality Scanner

**Protex™**
- BOM Service: Projects with used 3rd party components
- Clearing Service: Clearing Workflow and Tool Integration
- License Service: License Texts with Obligations Management
- Component Service: Catalogue of 3rd Party Components

**FOSSology**
- Catalogue of 3rd Party Components

**Projects with used 3rd party components**

**License Texts with Obligations Management**

**Catalogue of 3rd Party Components**

**External Tool Integration**
- Operated on Local Infrastructure with Access Control and Identity Management

**Developer**
- Quality Obligations
- Contacts
- Clearing Info
- License issues
- Vulnerabilities

**Project Manager**

**OS Officer**

**Sec Officer**
We are on Github!

www.github.com/sw360
Michael C. Jaeger  
Siemens AG Corporate Technology  
D-80200 Munich, Germany  
michael.c.jaeger@siemens.com

Johannes Kristan  
Bosch Software Innovations GmbH  
D-10785 Berlin, Germany  
johannes.kristan@bosch-si.com

Project site:  
https://github.com/sw360

Eclipse project information:  
https://projects.eclipse.org/projects/technology.sw360