DEPENDENCY-TRACK V4.13 PROJECT COLLECTIONS

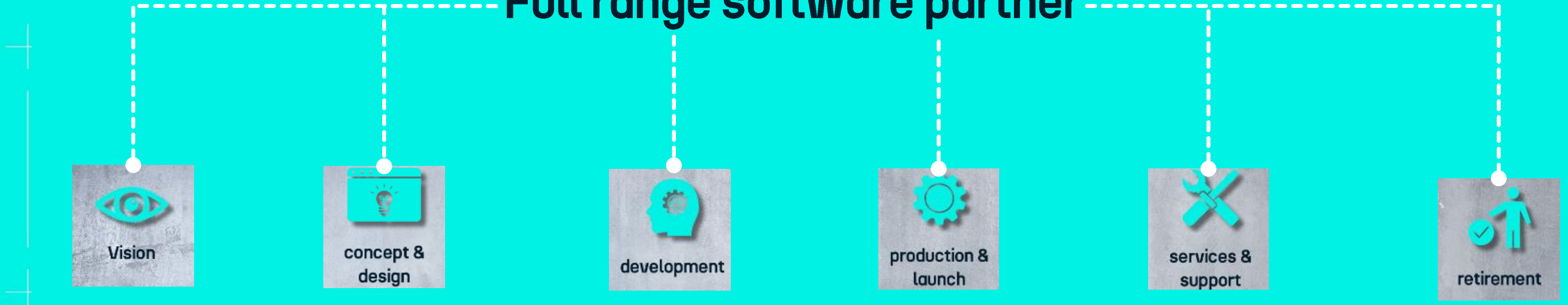
M&M software

Ralf King
Head of Competence Center
“Cyber- & Software-Security”

M&M SOFTWARE IN ONE SLIDE

M&M software

Full range software partner



~300
Employees

35+
Years experience in Software Development

4
Locations worldwide (2xGermany, China, India)

Industrial IT Security
TÜV SÜD
Secure Product Development Lifecycle

Secure Product Development Lifecycle assessed & monitored according to IEC 62443-4-1

PROJECT ORGANIZATION BEFORE V4.7

= Chaos in Dependency-Track

What we have:

- Many different customers
- Many different projects per customer
- Many different types of projects
  - Cloud deployments with multiple services and multiple environments (Prod/QA/Dev)
  - Versioned Software (Mobile Apps, Desktop clients, IoT Software, ...) with many different versions to track
- Many different teams working on different projects

What Dependency-Track had:

- A flat list of projects (= one single service/application version)
- No structure
  - What belongs together to one logical "project"?
  - Which (cloud-)environment does it belong to?
- Notifications configured per Dependency-Track project (or globally)
- Policies configured per Dependency-Track project (or globally)

PROJECT ORGANIZATION FROM V4.7

= get some structure in Dependency-Track

Part 1 of our solution:

- UI support for Parent/Child-relations
- Child-project support in alert rules
- Child-project support in policies

What we achieved:

- We could structure e.g. per customer > project > environment
- We could put all different project versions into one parent
- Much better overview ☺
- We could configure single alerts & policies high in the structure, auto-applying them to all children ☺

What the drawbacks were:

- Users were confused ☹:
  - Clicking on a parent did result in displaying an "empty" looking project
  - Numbers in project list did show 0 issues for all parents
- Users requested better overview/summary for higher-level entries (e.g. summary of issues of all 10 micro services in the PROD environment)

M&M software

PROJECT ORGANIZATION WITH V4.13

= much better usability & overview in Dependency-Track

Part 2 of our solution:

» Introduce isLatest version flag for projects (v4.12)
» Introduction of Collection Projects (v4.13)

What does it do?

» Define “structural/meta”-projects to be collections

» Collections do not have own components/vulns/policy violations

» Collections show contained children if opened

» Collections can aggregate metrics from children

- › All direct children
- › All direct children with specific tag
- › All direct children marked as latest

» Collections can be nested

What are example use cases?

» All versions of 1 application into one collection, showing only the metrics of the latest dev-version to quickly see not addressed issues

» Track & summarize metrics of multi-service cloud projects per environment (PROD/QA/DEV)

- › Put environments into another collection, highlight e.g. latest DEV to see not yet addressed issues, or PROD to quickly see current prod impact.

» Combine all above & summarize metrics of multi-application projects (Mobile App + Desktop client + Web Frontend + Cloud Backend + IoT App)

M&M software

WE TURN VISIONS FOR A DIGITAL WORLD INTO REALITY

WWW.MM-SOFTWARE.COM