

The diagram illustrates three sets of data represented as 10x10 grids:

- Component:** A 10x10 grid with a light orange background. Most cells are green. A few cells are highlighted in pink, indicating specific components of interest.
- Vulnerability:** A 10x10 grid with a light orange background. Most cells are green. One cell in the top-left corner is highlighted in yellow, indicating a specific vulnerability.
- VulnerableSoftware:** A 10x10 grid with a green background. All cells are green, indicating that all software in this set is vulnerable.

The figure displays four 8x8 grids, each representing a different entity type: Project, ServiceComponent, Cpe, and ICpe. The grids are color-coded to show the distribution of these entities across a spatial or categorical space. The Project and ServiceComponent grids feature yellow and pink cells, while the Cpe and ICpe grids feature green cells. The Project and ServiceComponent grids also show some cells with a small black dot in the top-left corner.

NotificationRule

ProjectProperty

AnalysisComment

Finding

PolicyTest

PolicyConfirmationTest

LicenseGroupTest

CweTest

TagTest

UniversityMainClass

The figure displays a 4x4 grid of resource tests. Each test is represented by a 4x4 grid of colored squares (green and yellow) indicating different states or metrics.

Test Name	Row 1	Row 2	Row 3	Row 4
UserResourceAuthenticatedTest	Green, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow
VulnerabilityResourceTest	Green, Green, Green, Green	Green, Green, Green, Green	Green, Green, Green, Green	Green, Green, Green, Green
OidcResourceAuthenticatedTest	Green, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow
PermissionResourceTest	Green, Green, Yellow, Green	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow
ProjectResourceTest	Green, Green, Green, Green	Green, Green, Green, Green	Green, Green, Green, Green	Green, Green, Green, Green
NotificationRuleResourceTest	Green, Green, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow
UserResource	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow	Yellow, Yellow, Yellow, Yellow

org.dependencytrack.persistence

QueryManager

PolicyQueryManager

ProjectQueryManager

VulnerabilityQueryManager

ComponentQueryManager

MetricsQueryManager

DefaultObjectGenerator

ServiceComponentQueryManager

FindingsQueryManager

VulnerableSoftwareQueryManager

RepositoryQueryManager

LicenseQueryManager

CweImporter

NotificationQueryManager

DefaultObjectGeneratorTest

CacheQueryManager

PackageMetadataProvider

The diagram illustrates the layout of the Java SE 8 JRE, organized into several main sections, each containing a grid of colored squares (yellow, green, purple) representing different components or files:

- IndexManager**: Contains a grid of 10 squares (3 yellow, 3 green, 4 purple).
- ComponentIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- ProjectIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- ServiceComponentIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- VulnerabilityIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- CpeIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- VulnerabilityComponentIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).
- SearchManager**: Contains a grid of 10 squares (3 yellow, 3 green, 4 purple).
- LicenseIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- VulnerabilityComponentIndexer**: Contains a grid of 6 squares (2 yellow, 2 green, 2 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).
- VulnerabilityIndexer (large)**: Contains a grid of 12 squares (4 yellow, 4 green, 4 purple).

Figure 1 displays a collection of 14 icons representing different software analysis tools. The icons are arranged in a grid-like fashion. Most are orange with white text and icons, while the 'MetaModel' icon is green with white text and icons. The tools shown are: NugetMetaAnalyzer, GoModulesMetaAnalyzer, CargoMetaAnalyzer, GemMetaAnalyzer, HexMetaAnalyzer, PypiMetaAnalyzer, RStudioMetaAnalyzer, ComposerMetaAnalyzer, MetaModel, NpmMetaAnalyzer, and two unlabeled icons (one orange, one pink). Each icon contains a small diagram representing the tool's architecture or data flow.

The screenshot shows the website [org.dependencytrack.ui](https://org.dependencytrack.ui). The page features a grid of tiles, each representing a different component and its version. The tiles are color-coded: green for 'ComponentVersion', 'DateUtil', 'HashUtil', 'DateUtilTest', 'HttpUtil', 'XmlUtil', and 'JsonUtil'; orange for 'HttpUtil', 'PurlUtil', and 'HttpUtilTest'; and a large green tile for 'GitHubSecurity'. Each tile displays a grid of small squares, likely representing a version matrix or a list of versions. The 'GitHubSecurity' tile is the largest and is located on the right side of the page.

[illegible]

The diagram illustrates a distributed system architecture for a policy engine. It is organized into three main sections: a left column of test components, a central cluster of evaluation and engine components, and a right column of engine and test components.

- Left Column (Green Background):**
  - CoordinatorPolicyEvaluatorTest:** A large green rectangle containing a 4x4 grid of smaller green squares. The bottom-right square is labeled "CoordinatorPolicyEvaluator".
  - VersionPolicyEvaluatorTest:** A green rectangle containing a 1x4 grid of smaller green squares.
- Central Cluster (Orange Background):**
  - CoordinatorPolicyEvaluatorTest:** An orange rectangle containing a 1x4 grid of smaller orange squares.
  - VersionPolicyEvaluatorTest:** An orange rectangle containing a 1x4 grid of smaller orange squares.
  - PolicyEngine:** An orange rectangle containing a 2x2 grid of smaller orange squares. The bottom-right square is labeled "PolicyEngine".
  - VersionPolicyEvaluatorTest:** An orange rectangle containing a 1x4 grid of smaller orange squares.
- Right Column (Orange Background):**
  - PolicyEngine:** An orange rectangle containing a 2x2 grid of smaller orange squares. The bottom-right square is labeled "PolicyEngine".
  - VersionPolicyEvaluatorTest:** An orange rectangle containing a 1x4 grid of smaller orange squares.

Connections are indicated by lines between the components, showing a network topology where the test components on the left interact with the evaluation and engine components in the center and right.

[illegible]

The diagram shows a flow from a large green box labeled 'Advisory' to a yellow box labeled 'AnalysisRequest'. The 'Advisory' box contains a 5x5 grid of smaller green boxes, with the bottom row having only four boxes. Below this grid is a green box labeled 'AdvisoryResults'. The 'AnalysisRequest' box contains a row of five yellow boxes, each with a small green box inside. Below this row are four orange boxes: 'AnalysisRequestProcessor', 'AnalysisRequestProcessor', 'AnalysisRequestProcessor', and 'AnalysisRequestProcessor'. Each orange box contains a row of four yellow boxes, each with a small green box inside. Below these orange boxes are four yellow boxes: 'AnalysisRequestProcessor', 'AnalysisRequestProcessor', 'AnalysisRequestProcessor', and 'AnalysisRequestProcessor'. Each yellow box contains a row of four yellow boxes, each with a small green box inside.

The figure displays three wireframe diagrams of a dashboard layout, each with a different color scheme (green, orange, and blue). Each diagram is divided into several sections, including a 'Header', 'Footer', and a main content area. The main content area is further divided into sub-sections, some of which contain placeholder text or icons. The diagrams illustrate different ways to organize and present information on a dashboard.

```

graph TD
    Root[ ]
    Root --- KSU[KernelSecurityUploader]
    Root --- MC[ModelConverter]
    Root --- NP[NvdParser]
    KSU --- KSUT[KernelSecurityUploaderTest]
    MC --- MC_Exception[org.dependencycheck.parser.common.exception]
    MC --- MC_Mitre[org.dependencycheck.org.mitre]
    NP --- NP_Nvd[org.dependencycheck.parser.nvd]
    MC_Exception --- MC_Exception_Test[ClassNotFoundErrorTest]
    MC_Mitre --- MC_Mitre_Test[NvdParserTest]
    NP_Nvd --- NP_Nvd_Test[NvdParserTest]
  
```

[illegible]