Eclipse Arrowhead

Roadmap WG

ARROWHEAD

Roadmap WG Agenda - MoM 250508

1. Updates to the GSoSD, Jerker
Support systems will be added in same document first with SysML models based on Arrowhead DSL.

2. Core system development status, AITIA
Documentation and code is available here: https://aitia-iiot.github.io/ah5-docs-java-spring/home/welcome/

3. Core systems implemented in Golang, Dezyne strategy
see slide 3. Discussion from Johan on how tests can be used to provide robust code implementation. Was concluded that Johan and Tamas/Raymond should have deeper lock into how this can introduced into our CI/CD process. The possibilities of the naming convention to enable a security issue in relation to the x.509 certificate was rised. No immediate security risk was identified.

4. Naming convention updates, Tamas, Raymond, Jan vD, and Jerker
see slide 4-5

5. Data model translation/integration based on ontologies - Arrowhead Framework Ontology, Jan vD
Jan show how the Arrowhead Framework Ontology can be used to enable reasoning on the current operational state is compliance with desired state.

6. How to provide identity?
Pushed to next meeting

7. AITIA v5.0 to make a feature presentation at the bi-weekly meetings.
Scheduled for May 20

8. Backward compatibility, AITIA
Slightly discussed in connection with item 3. Will be addressed when we are closer to first release of v.5

7. Issues in GitHub as time permit
Pushed to next meeting

8. Next meeting:
Friday May 23, 8.30-10.00

Core system implementation

Java - reference implementation

limitations are memory hungry, speed

Golang

Early versions of core systems exists

slim on memory, fast

Dezyne

Support for code generation and formal verification of generated executable code, C++, Pyton, Java script

SysML v2

Support for MBSE

Arrowhead DSL (now in SysML v1.6). Translation

Naming convention

Why naming convention is necessary?

- System, service, operation, device and interface names are used in programing languages, URLs, DNS entries, file systems.
  - → Only ASCII characters to be allowed
- Sometime system name and service name very similar (even the same), but people should be able to differentiate easiliy by reading.
- We should promote consistency in naming within the community
  - → To use different styling per entity
    - System name: PascalCase (ServiceRegistry)
    - Service name: camelCase (serviceDiscovery)
    - Service operation name: kebab-case (get-entries)
    - Interface name: snake_case: generic_http
    - Device name: UPPER_SNAKE_CASE (MY_DEVICE)
  - → To apply convention also to composite identifiers
    - Service instance identifiers: SystemName<*delimiter*>serviceName<*delimiter*>version (ServiceRegistry<*delimiter*>serviceDiscovery<*delimiter*>1.0.0)
      - → What to use as delimiter? ("|" is proposed)
    - Cloud identifiers: CloudName<*delimiter*>Organization (TestCloud<*delimiter*>AitiaInc)
      - → What to use as delimiter? ("|" is proposed)

How to align with other standards in use?

- URL
  Reserved characters: ! * ' ( ) ; : @ & = + $, / ? # [ ]
  - → Not to use these as separator in composite identifiers.
- Certificate authentication method:
  X.509 certificate common name allows only: uppercase letters, lowercase letters, digits, hyphen (but not at the start or end) and dot (as a separator between domain levels)
  - → To use kebab-case in certificates and transform in code level
    (service-registry.test-cloud.aitia-inc.arrowhead.eu = ServiceRegistry.TestCloud.AitiaInc.arrowhead.eu)
  - → Not to use dot as separator in composite identifiers.

  CN represents a fully qualified domain name, which can be maximum 253 character and max 63 char per label. (subdomain.example.com)
  - → To apply max 63 char rule to the names
- Semantic versioning: <major>.<minor>.<patch>
  - → Not to use dot as separator in composite identifiers.
- RDF (Resource Description Framework) for Knowledge Graphs:
  Reserved characters: <, >, &, ",
  - → Not to use these as separator in composite identifiers.

camelCase is also recommended to use in attributes (keys in payloads, metadata, properties)!

Naming convention

How to align with other standards in use?

- URL
  Reserved characters: ! * ' ( ) ; : @ & = + $, / ? # [ ]

  → Not to use these as separator in composite identifiers.

- Certificate authentication method:
  X.509 certificate common name allows only: uppercase letters, lowercase letters, digits, hyphen (but not at the start or end) and dot (as a separator between domain levels)

  → To use kebab-case in certificates and transform in code level
  (service-registry.test-cloud.aitia-inc.arrowhead.eu = ServiceRegistry.TestCloud.AitiaInc.arrowhead.eu)

  → Not to use dot as separator in composite identifiers.

  CN represents a fully qualified domain name, which can be maximum 253 character and max 63 char per label.
  (subdomain.example.com)

  → To apply max 63 char rule to the names

- Semantic versioning: <major>.<minor>.<patch>

  → Not to use dot as separator in composite identifiers.

- RDF (Resource Description Framework) for Knowledge Graphs:
  Reserved characters: <, >, &, “,

  → Not to use these as separator in composite identifiers.

camelCase is also recommended to use in attributes (keys in payloads, metadata, properties)!