Eclipse Arrowhead

-

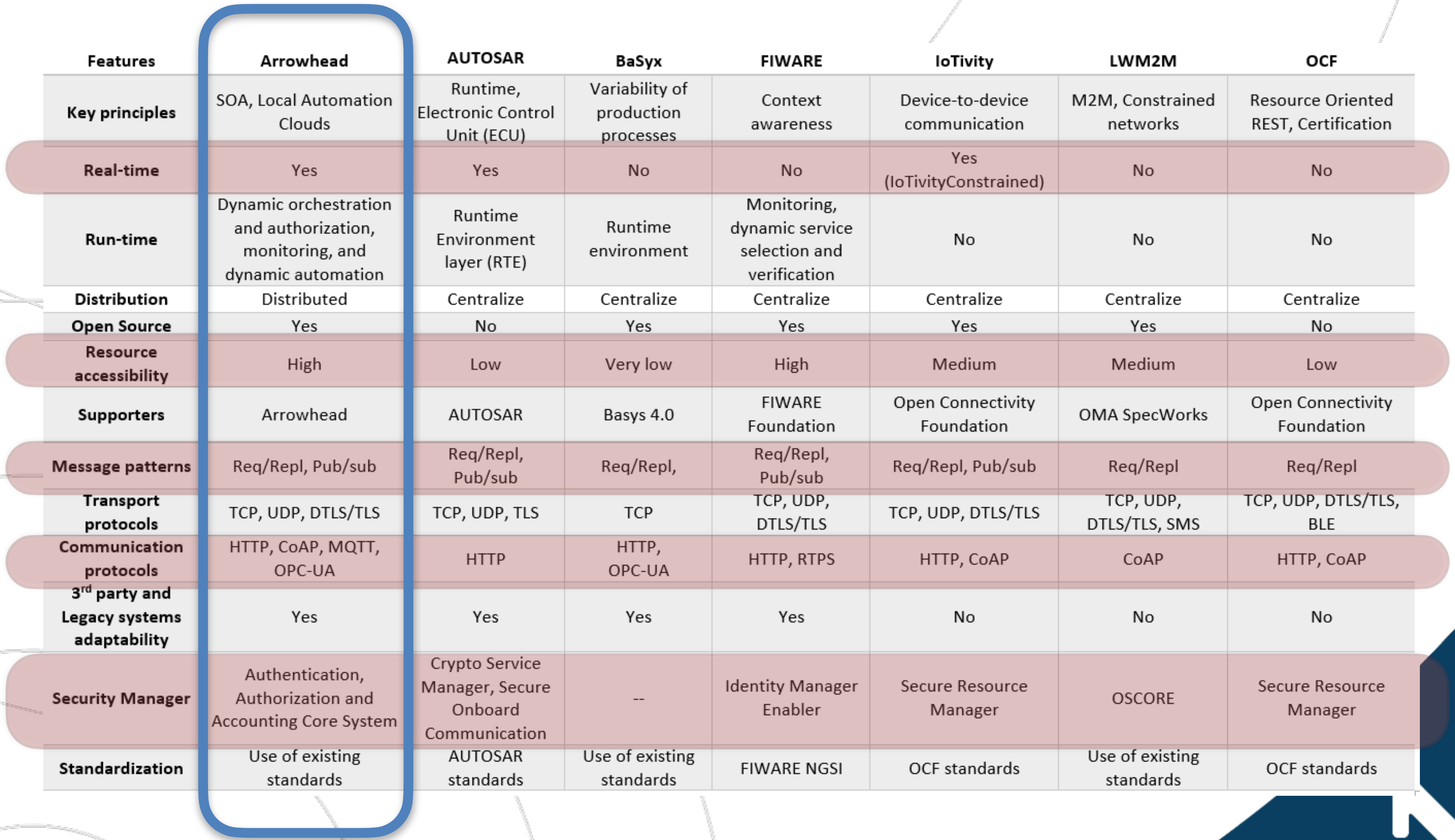
basic principles of the framework

ARROWHEAD TOOLS

Arrowhead framework

- System of Systems
  - Interoperabity, Integrability, Independence
  - Service Oriented Architecture
- Local Automation Clouds
- Varios Multi-clouds: Edge, Platform, Enterprise
- Maturity Levels for Integrability – hardware and software adaptors
- Mandatory and Supporting Core Systems
- Collaborations with other Eclipse projects

A comparison of IoT-SoS Architectures & Platforms



Features	Arrowhead	AUTOSAR	BaSyx	FIWARE	IoTivity	LWM2M	OCF
Key principles	SOA, Local Automation Clouds	Runtime, Electronic Control Unit (ECU)	Variability of production processes	Context awareness	Device-to-device communication	M2M, Constrained networks	Resource Oriented REST, Certification
Real-time	Yes	Yes	No	No	Yes (IoTivityConstrained)	No	No
Run-time	Dynamic orchestration and authorization, monitoring, and dynamic automation	Runtime Environment layer (RTE)	Runtime environment	Monitoring, dynamic service selection and verification	No	No	No
Distribution	Distributed	Centralize	Centralize	Centralize	Centralize	Centralize	Centralize
Open Source	Yes	No	Yes	Yes	Yes	Yes	No
Resource accessibility	High	Low	Very low	High	Medium	Medium	Low
Supporters	Arrowhead	AUTOSAR	Basys 4.0	FIWARE Foundation	Open Connectivity Foundation	OMA SpecWorks	Open Connectivity Foundation
Message patterns	Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl,	Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl	Req/Repl
Transport protocols	TCP, UDP, DTLS/TLS	TCP, UDP, TLS	TCP	TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS, SMS	TCP, UDP, DTLS/TLS, BLE
Communication protocols	HTTP, CoAP, MQTT, OPC-UA	HTTP	HTTP, OPC-UA	HTTP, RTPS	HTTP, CoAP	CoAP	HTTP, CoAP
3rd party and Legacy systems adaptability	Yes	Yes	Yes	Yes	No	No	No
Security Manager	Authentication, Authorization and Accounting Core System	Crypto Service Manager, Secure Onboard Communication	--	Identity Manager Enabler	Secure Resource Manager	OSCORE	Secure Resource Manager
Standardization	Use of existing standards	AUTOSAR standards	Use of existing standards	FIWARE NGSI	OCF standards	Use of existing standards	OCF standards

C. Paniagua and J. Delsing, "Industrial Frameworks for Internet of Things: A Survey," in *IEEE Systems Journal*, doi: 10.1109/JSYST.2020.2993323.

Interoperbility & Integrability & Indepencence

Arrowhead Objectives

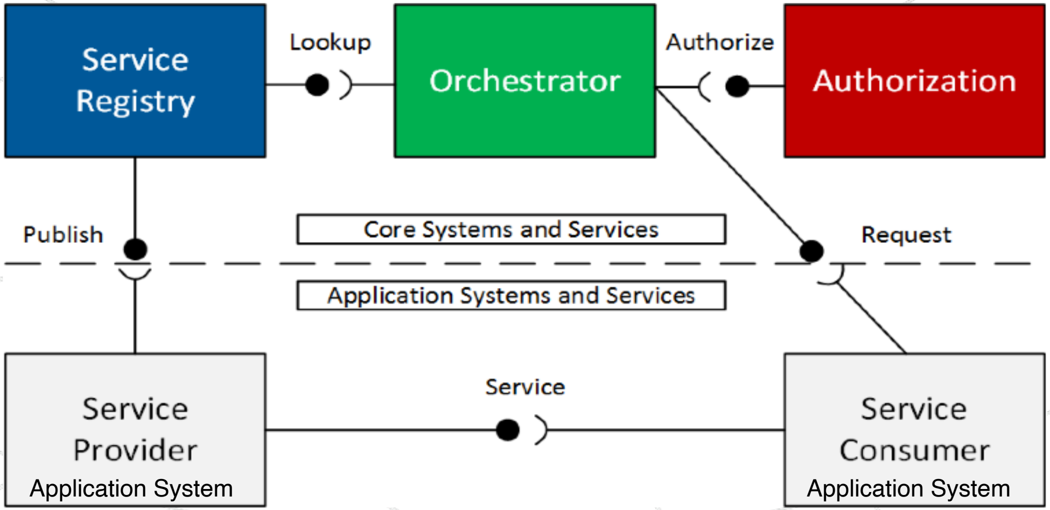
- **System of Systems for Industrial IoT and CPS**

- **Interoperability**: Service Oriented Architecture
  - Late Binding - Loose Coupling – Lookup (of Service Consumers+Providers)
- **Integrability**
  - easy interaction between Legacy and New (native Arrowhead) systems
- **Independence**
  - from underlying technologies (services)
  - from application protocols (translation)

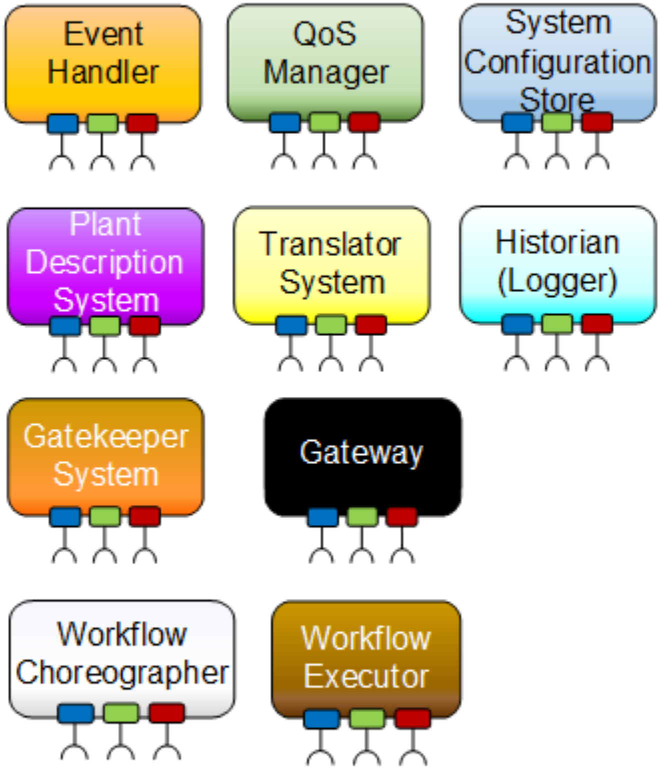
ARROWHEAD TOOLS

Core Systems
in the Arrowhead Service Oriented Architecture

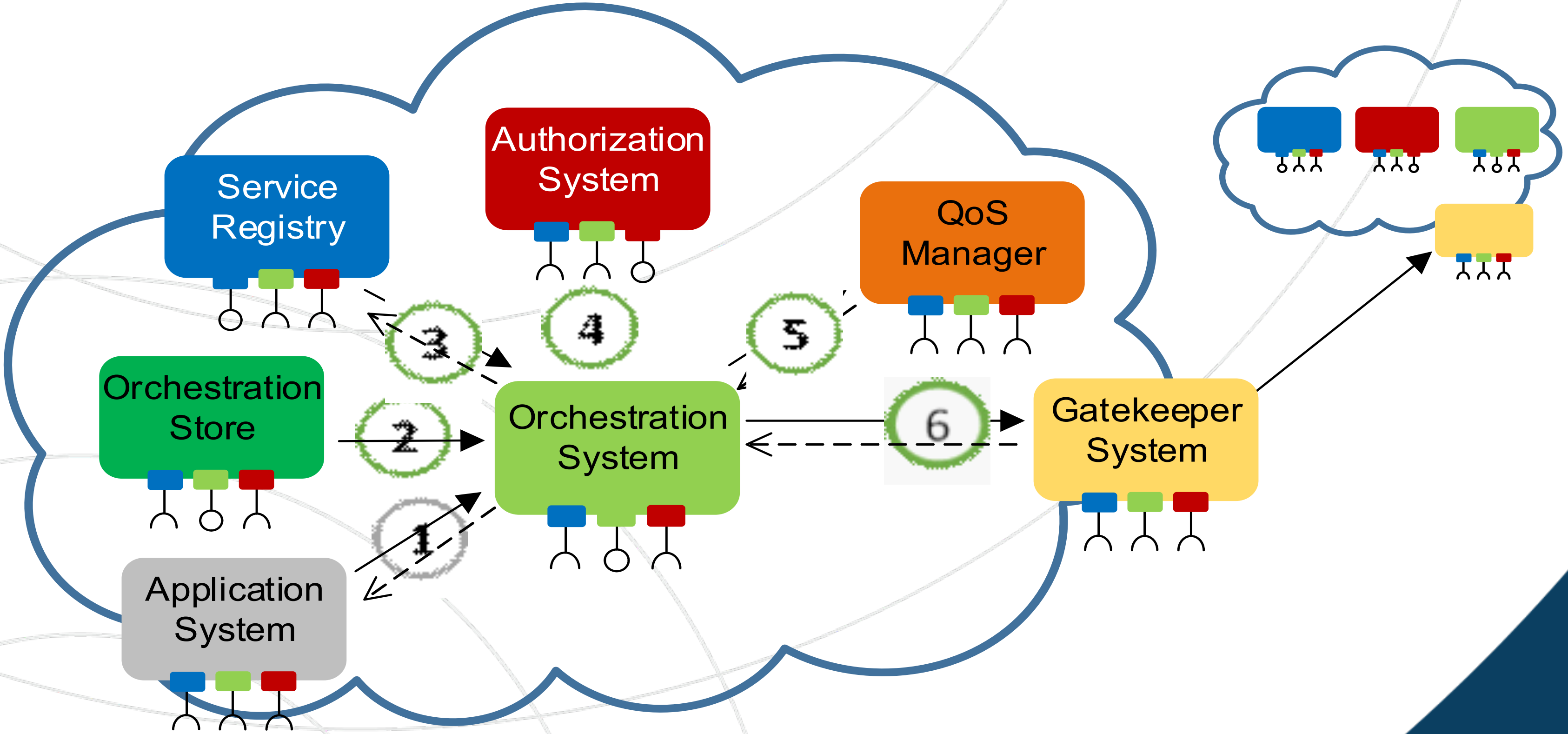
Mandatory Core Systems



Supporting Core Systems



Orchestration: finding a fitting provider for a service consumer

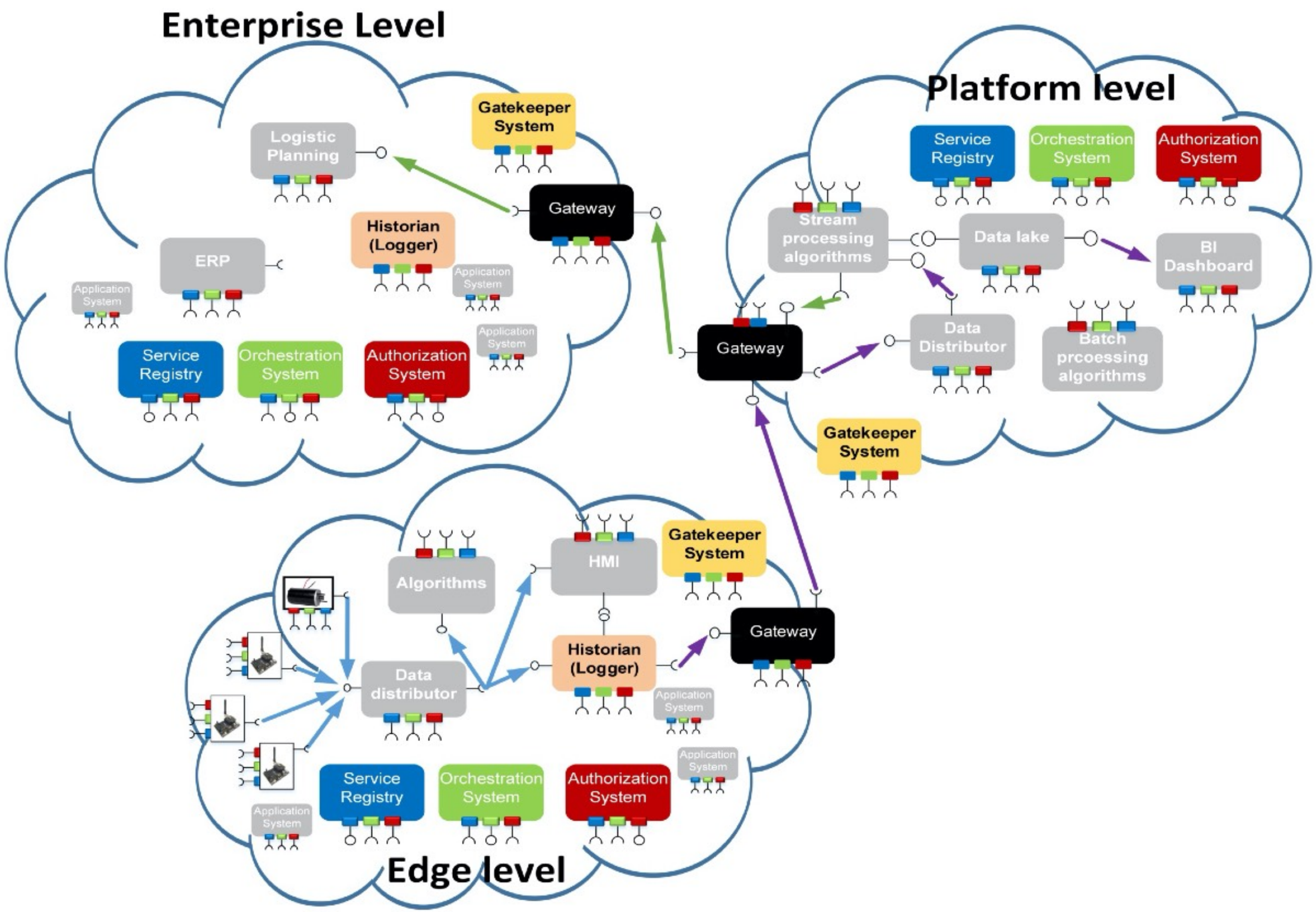


Either given in the OrchestrationStore,

or choosing from the ServiceRegistry,

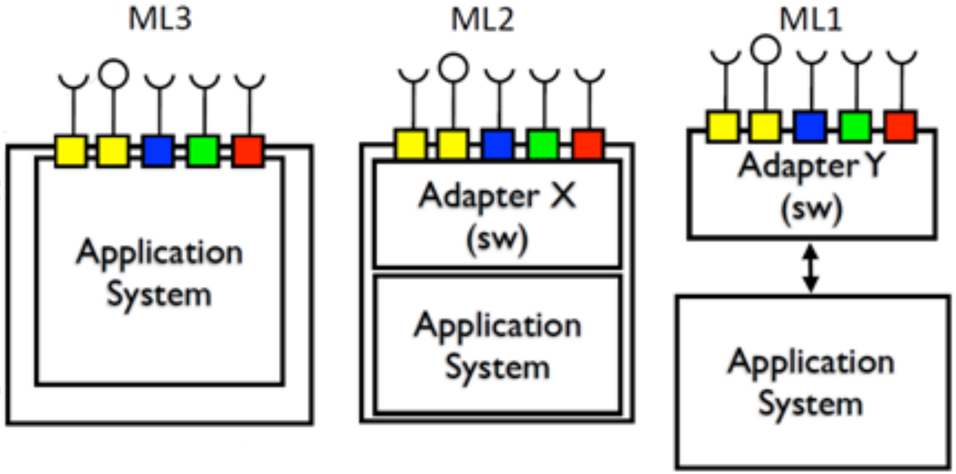
…or Global Service Discovery with Inter-Cloud orchestration

Multi-level Service Exchange



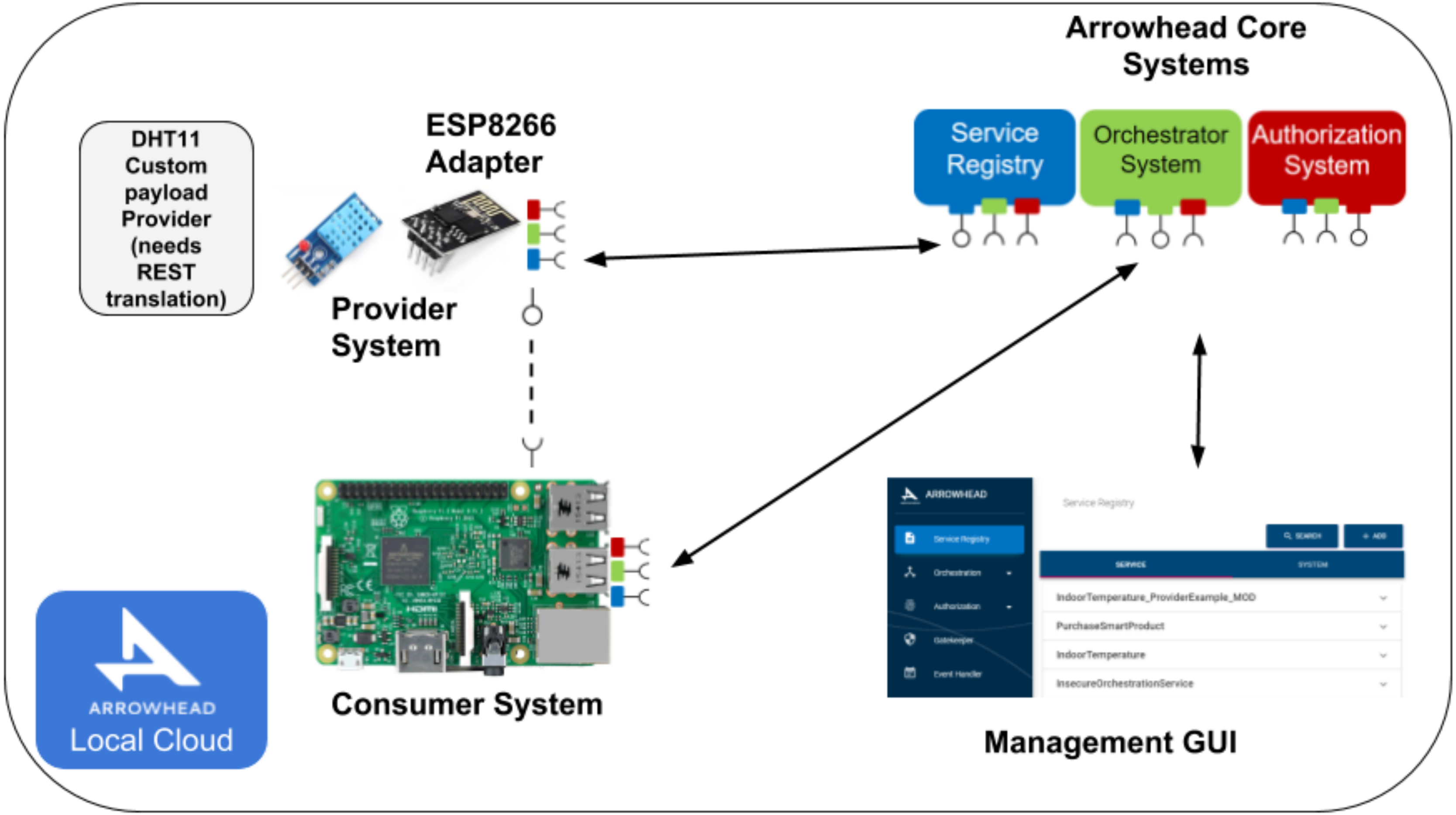
ARROWHEAD TOOLS

Maturity Levels of Arrowhead Integration



- Native Arrowhead capabilities (ML3)
- Software adapters (ML2)
- Hardware adapters (ML1)

An example for „Maturity Level 1” provider



IndoorTemperature_ProviderExample_MOD

PurchaseSmartProduct

IndoorTemperature

InsecureOrchestrationService

Examples of supporting core system features

- **Translator:**

to allow protocol translation between various application systems (e.g. RESTful HTTP, CoAP, MQTT, OPC-UA, etc.)

- **OnboardingController:**

handle device onboarding to the local cloud with the support of SystemRegistry and DeviceRegistry

- **Choreographer:**

to manage workflows and execute recipes in a dynamic manner: the next step is executed depending on service provider availability

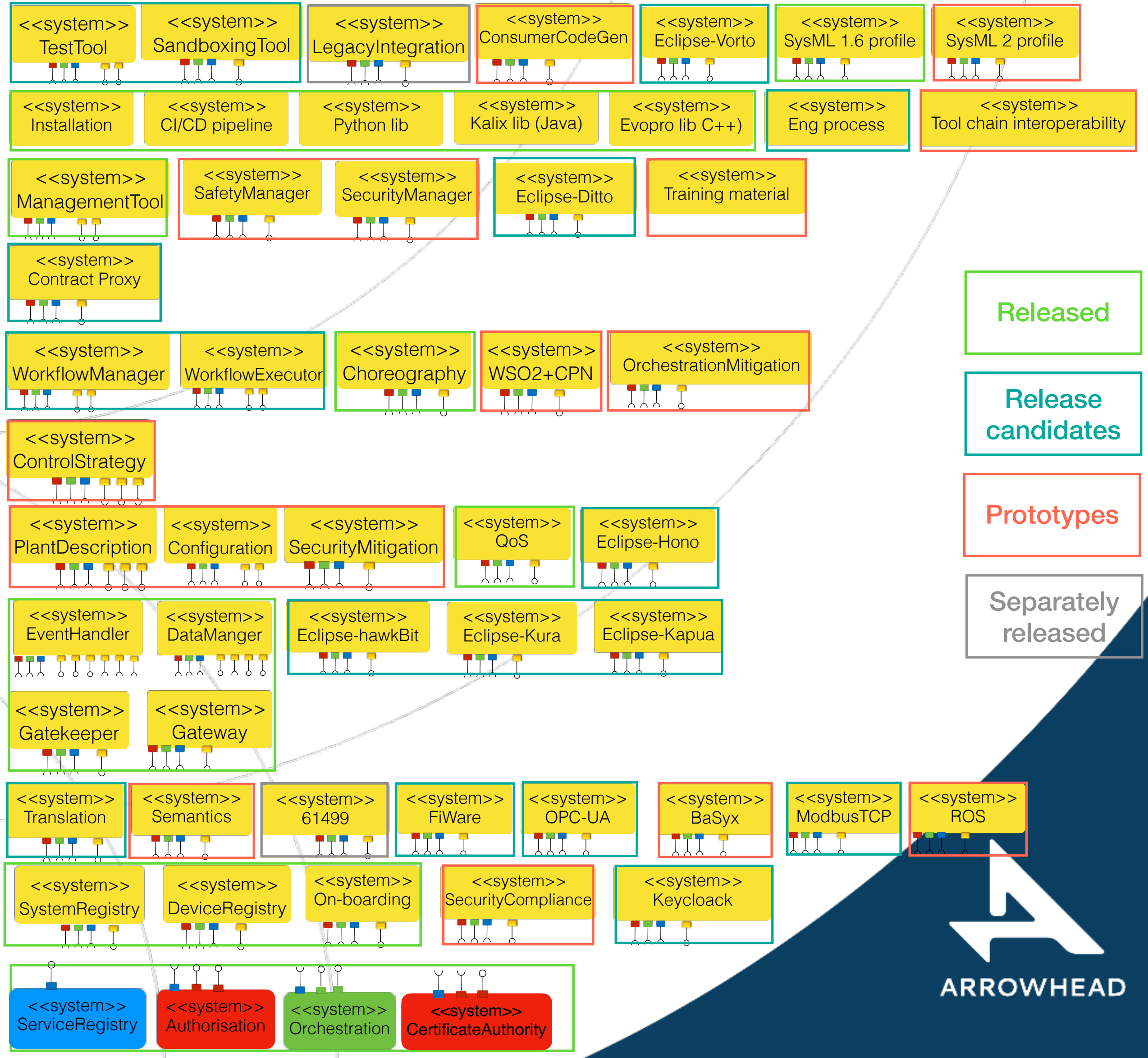
- **PlantDescription:**

to provide up-to-date information on the engineering plant

- **... Event Handler, QoS Manager, CertificateAuthority, etc.**

ARROWHEAD TOOLS

Arrowhead v4.2.0



Engineering tools

<<system>> TestTool
<<system>> SandboxingTool
<<system>> LegacyIntegration
<<system>> ConsumerCodeGen
<<system>> Eclipse-Vorto
<<system>> SysML 1.6 profile
<<system>> SysML 2 profile
<<system>> Installation
<<system>> CI/CD pipeline
<<system>> Python lib
<<system>> Kalix lib (Java)
<<system>> Evopro lib C++)
<<system>> Eng process
<<system>> Tool chain interoperability

Management support:

<<system>> ManagementTool
<<system>> SafetyManager
<<system>> SecurityManager
<<system>> Eclipse-Ditto
<<system>> Training material

Supply chain/product life cycle

<<system>> Contract Proxy

Execution support

<<system>> WorkflowManager
<<system>> WorkflowExecutor
<<system>> Choreography
<<system>> WSO2+CPN
<<system>> OrchestrationMitigation

Control support

<<system>> ControlStrategy

System of Systems support

<<system>> PlantDescription
<<system>> Configuration
<<system>> SecurityMitigation
<<system>> QoS
<<system>> Eclipse-Hono
<<system>> EventHandler
<<system>> DataManger
<<system>> Eclipse-hawkBit
<<system>> Eclipse-Kura
<<system>> Eclipse-Kapua

Inter cloud service exchange

<<system>> Gatekeeper
<<system>> Gateway

Interoperability

<<system>> Translation
<<system>> Semantics
<<system>> 61499
<<system>> FiWare
<<system>> OPC-UA
<<system>> BaSyx
<<system>> ModbusTCP
<<system>> ROS

Secure infrastructure:

<<system>> SystemRegistry
<<system>> DeviceRegistry
<<system>> On-boarding
<<system>> SecurityCompliance
<<system>> Keycloack

Local cloud basic properties:

<<system>> ServiceRegistry
<<system>> Authorisation
<<system>> Orchestration
<<system>> CertificateAuthority

Released

Release candidates

Prototypes

Separately released

ARROWHEAD

Interactions with other Eclipse Projects

Already active interaction

- **4DIAC -** PLC framework based on IEC 61499 standard
- **Vorto -** IoT device capability descriptions, catalogues
- **HawkBit -** IoT device management
- **Ditto -** Digital Twin state management and more
- **Hono -** Protocol-independent communication with the IoT backend solution
- **Paho -** MQTT and MQTT-SN implementations in various languages
- **Mosquitto -** Lightweight MQTT Server implementation

Collaboration with involved partners

- **Kapua -** Connect, manage and integrate IoT devices and their data through MQTT
- **Kura -** API access to IoT Gateway HW interfaces and provide MQTT connectivity

Leshan - OMA Lightweight M2M server and client in Java (LWM2M)

- **Wakaama -** OMA Lightweight M2M C implementation (LWM2M)
- **Keti -** to protect RESTful API using Attribute Based Access Control (ABAC).
- **Milo, Unide, OM2M, etc.**

ARROWHEAD TOOLS

Brief summary - Arrowhead framework

- System of Systems
  - Interoperabity, Integrability, Independence
  - Service Oriented Architecture
- Local Automation Clouds
- Varios Multi-clouds: Edge, Platform, Enterprise
- Maturity Levels for Integrability – hardware and software adaptors
- Mandatory and Supporting Core Systems
- Collaborations with other Eclipse projects