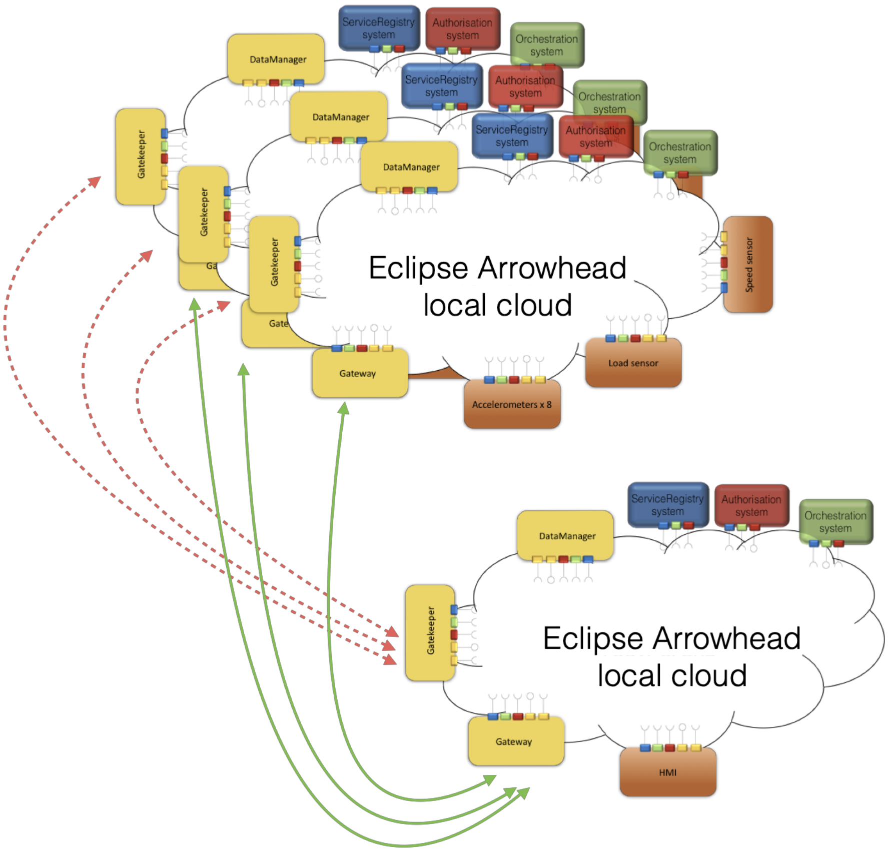
Fundational principles

Eclipse Arrowhead



Abstract
This document captures the foundational ideas and prinicples for the Eclipse Arrowhead architecture and implementation platform.

Contents

1 Overview

This document describes the fundational principels for Eclispe Arrowhead architecture and implementation platform.

1.1 Significant Prior Art

Eclipse Arrowhead has its roots in service oriented arcitecture, SOA, and its use for primarily far edge, edge and fog automation and digitalsiation with interoperability to the cloud level.

A set of EU projects have built the foundation for whats now Eclipse Arrowhead, current version 4.6.1 with the specifications for v5.0 in the works. These projects are:

- Socrades
- IMC-AESOP
- Arrowhead
- Productive4.0
- Arrowhead Tools
- AIMS5.0
- Arrowhead fPVN

A couple of basic architectture ideas has been around since the prior art projects:

- Socrades:
  - Hard real time control using internet protocols
- IMC-AESOP:
  - Objective to be capable of implementing real world SCADA and DCS systems.
  - The local cloud concept was born. Local clouds are self contined for its intended operation enabling local security, protection and if equiped with TDMA network MAC real time properties can be achived.
  - System are self contained for its intended operation, e.g. owning and responsible for its own data storage and compitational resources.
  - Arrowhead:
    - Objective to be interoperability to legacy and internet protocols and being Open Source.
    - Basic SOA foundation established, Look-up, Late binding and Lossely coupled.
    - Mandatory core systems defined: ServiceRegistry, Orcehstration, Authorisation
    - Interoperability enabled through translation dynamically instatiated when needed.
    - v3.3 released as open source
  - Productive4.0:
    - Arrowhead Framework becomes Eclipse Arrowhead architecture and implementation platform
    - Extending the implementation platform - teh Arrowhead technology stack is defined
    - v4.5 released
  - Arowheead Tools
    - Objective to reduce engineereing cost with 20-50%
    - Extending the Eclipse Arrowehad technology stack
    - v4.6 released
    - Achived 30-95%engineering cost and time reduction in 28 industrial use cases along the extended IEC 81346 engineering process.

In summary the industrial requirements setting hte scene for the Eclipse Arrowhead developments comes from industiral automation and digitalisation related to domains like e.g. automotive, aeronatics, manufacturing, batch and continous processing, semiconductor production, maintenance, buildings, energy, mining, logitics and smart cities. The primary focus for the Eclispe Arrowhead SoS view is integration and interoperability of edge and deep edge technology into System of Systems with capabilites to connect to cloud e.g. for data storage and high performance computing.

The current comprehensive high level architecture description of Eclipse Arrowhead architecture is the book "IoT Automation - Arrowhead framework" [1]. The currently released core systems and associated documentations are avialable at `www.github.com/eclipsearrowhead`.

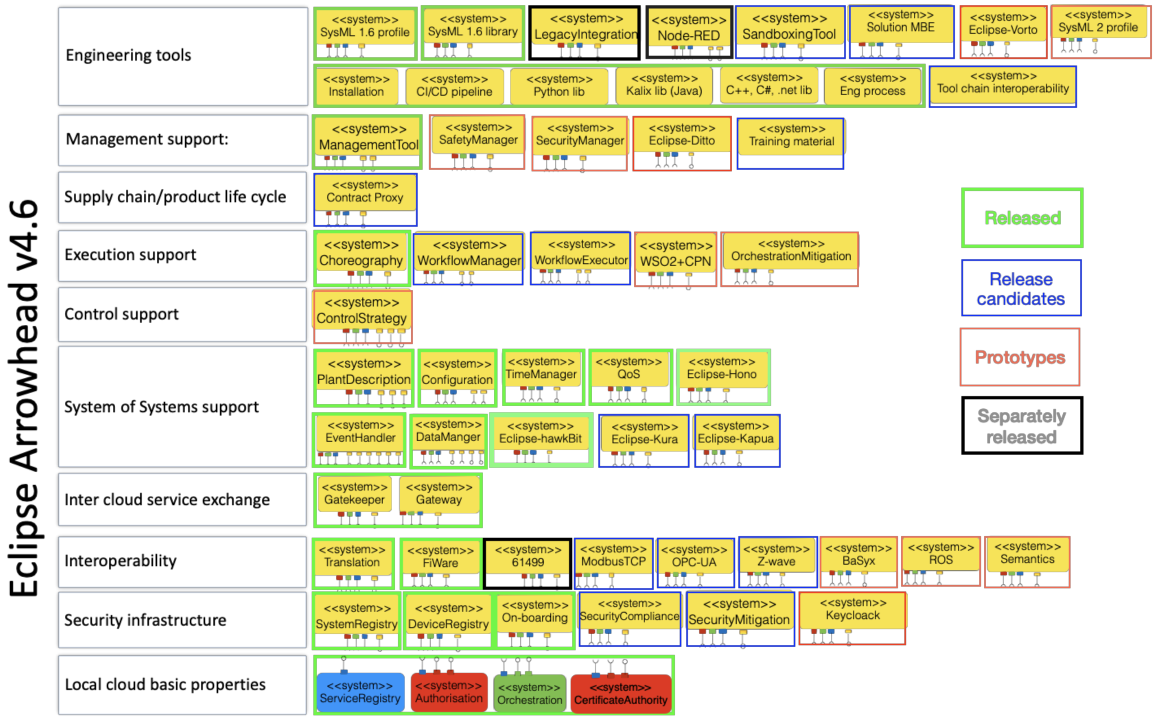


Figure 1: The Eclipse Arrowhead technology stack an dassoicated microsystems, released, relase candidates and protoypes.

1.2 Eclipse Arrowhead architecture philophosy

The architecture philiposy is based on the following key technology decission and objectives:

- Key technology decisions
  - A fully distributed microservice SOA approach shall be used
  - Support for design and run time engineering
  - A set of microsystems, the technology stack cf. Figure 1, shall be provided. Use of one or many of these microsystems enables the implementation of automation and digitalisation solutions.
    * Three core microsystems considered as primary and almost mandatory, ServiceRegistry, Orcehstration, Authorisation, enabling Look-up, Late binding and Lossely coupling.
    * A set of support microsystem will be defined and implemented covering the technology stack cf. Figure enabling implementation automation architectures like ISA-95 and RAMI4.0. 1.
  - Basic microsystem properties: A micosystem can be stateless or statefull. If statefull the microsystem is responsible for its own data stoarage perferable using a database and a well established/standardised datamodel.

- The local cloud concept shall be used providing segmentation and protection of functionall properties enabling differentieade security, safety, and real time properties within a solution architecture with managed access into a segment and between segements.
- Network technology agnostic, allowing for different network properties inside different local clouds.
- Documentation of the Eclipse Arrowhead architecture and solutions based thereon to follow the adopted documentation structure as shown in Figure 2.

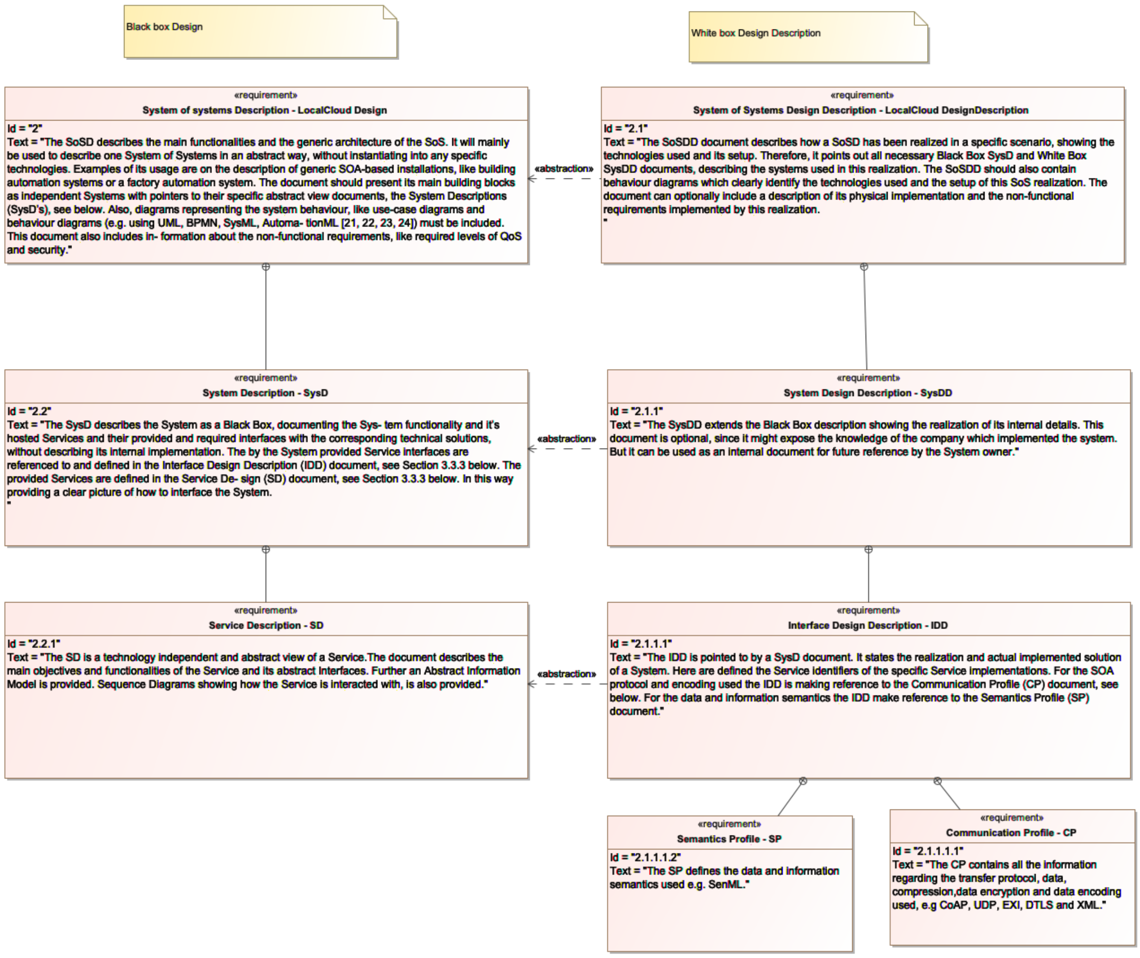


Figure 2: The Eclipse Arrowhead documentation structure.

- Architectural levels defined are depicted in Figure 3.

- Technology objectives:
  - Interoperability support at Service level shall be provided regrding: SOA, IP and legacy protocols, encodings, compressions, security, data models, trough translators or dedicated adaptors. For data model interopeability between major standards like e.g. ISO10303, ISO 15926, IEC 81346 are prioritised.
  - Security shall be supported at service exchange level with authentication, authorisation and audit. Security at finer granularity that service level is being addressed. Securityy is highly recommended but can be discarded if desired.
  - Secure on-boarding: On-boarding based on authentication of devices, microsystems and microservices shall be supproted.

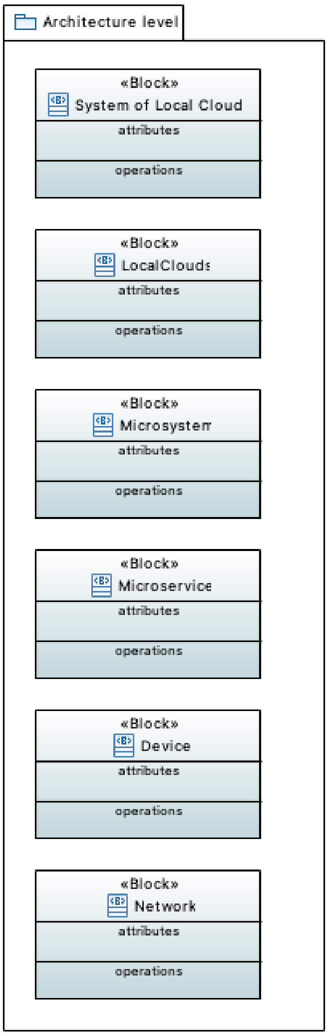


Figure 3: The Eclipse Arrowhead architectural levels from network to System of Systems.

– Support for multipe strategy direction. A strategy directions may be: Security, LifeCycle, Maintenance, Business, Audit, Monitoring, BusinessAdminstration, BusinessModels. This will also require ways of addressing interdependecies between the various stratgy directions.
– Model based engineering to support documentation, requirements validation, automated code generation and generation of deployment ready code packages (containers) [2, 3]
– Domain specific language based on UML/SysML [4].

I set of terminologies used in the Eclipse Arrowhead devlopements and supporting project are important to highlight. The following are important to higlight since they may have different meaning in other domains, contexts and projects. The document by Emanuel Palm and Jerker Delsing has defined the most important vocabulary within the Eclipse Arrowhead development [5]. Some important updates are:

- Service -> Microservice
- System -> Microsystem

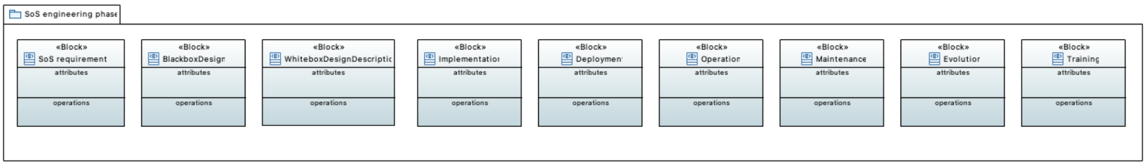


Figure 4: The Eclipse Arrowhead engineering process is an extension of the IEC81346 engineering process.

2 Eclipse Arrowhead engineering process

The Arrowhead engineering processs has primarly been documented in a couple of papers by G. Urgese et.al. [6, 7], cf. Figure 4.

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

Figure 5

3 Comparison to other frameworks

A recent comparison to other frameworks has been published by Cristina Paniagua et.al [8]. The summary table of this comparison is reproduced in Figure 5.

4 Implementation specifics

For the current implementations a set of technologies has been choosen for various reasons ranging from simple to use over "this are the technlogies I'm familiar to" to ther are the technologies "which provides what I what". Technologies used for implementations of Eclipse Arrowhead microsystem are:

- Protocols: HTTP (REST), CoAP, MQTT, Websocket
- Network protocols: Ethernet, WiFi, 802.15.4
- Datamode standards used: SenML

- Security protocol: X.509 certificates and tokens, see [9]
- Programing language: Java, Python
- Libraries: Java, C
- High performance: GO lang

5 References

[1] J. Delsing, Ed., *IoT Automation - Arrowhead Framework*. CRC Press, Feb. 2017, no. ISBN 9781498756754.

[2] J. Delsing, G. Kulcsár, and O. Haugen, "Sysml modeling of service-orientedsystem-of-systems," *Innovations in Systems and Software Engineering*, no. https://doi.org/10.1007/s11334-022-00455-5, 2021.

[3] J. Delsing, "Smart city solution engineering," *Smart Cities*, vol. 4, no. 2, pp. 643–661, 2021. [Online]. Available: https://www.mdpi.com/2624-6511/4/2/33

[4] ——, "Eclipse arrowhead dsl - sysml profile and uml meta model," https://github.com/eclipse-arrowhead/profile-library-sysml, Oct. 2022.

[5] E. Palm and J. Delsing, "Eclipse arrowhead concept reference," https://github.com/eclipse-arrowhead/documentation/blob/master/distribution/Eclipse%20Arrowhead%20Concepts%20Reference%20v0.3.Proposal.pdf, Feb. 2023.

[6] G. Urgese, P. Azzoni, J. v. Deventer, J. Delsing, and E. Macii, "An engineering process model for managing a digitalised life-cycle of products in the industry 4.0," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, April 2020, pp. 1–6.

[7] G. Urgese, P. Azzoni, J. van Deventer, J. Delsing, A. Macii, and E. Macii, "A soa-based engineering process model for the life cycle management of system-of-systems in industry 4.0," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7730

[8] C. Paniagua and J. Delsing, "Industrial frameworks for internet of things: A survey," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1149–1159, 2021.

[9] E. Palm, "Eclipse arrowhead x.509 certificate profiles, profile description (pd)," "https://github.com/eclipse-arrowhead/documentation/blob/master/distribution/EclipseArrowheadX.509CertificateProfilesv1.0.Proposal.pdf", Feb. 2021.

6 Revision History

6.1 Amendments

Revision history and Quality assurance as per examples below

No.	Date	Version	Subject of Amendments	Author
1	2023-05-08	4.6.1		Jerker Delsing
2				
3				

6.2 Quality Assurance

No.	Date	Version	Approved by
1	2022-01-10	4.6.1	