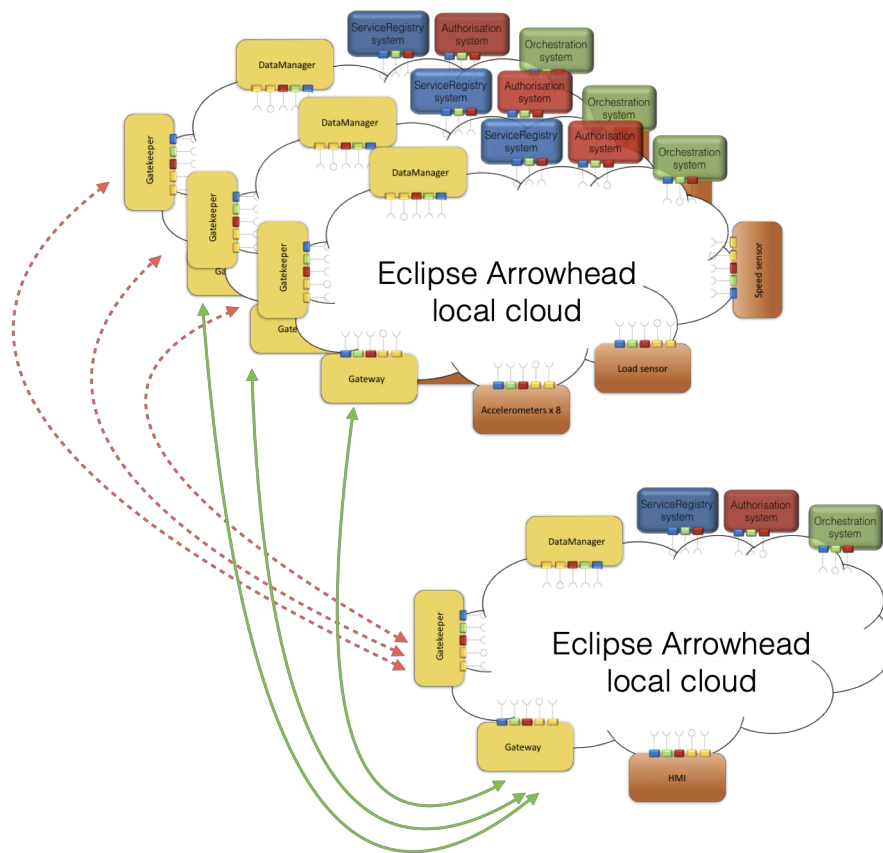


Fundational principles

Eclipse Arrowhead



Abstract

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



ARROWHEAD

Document title
Fundational principles
Date
2023-05-28

Version
4.6.1
Status
DRAFT
Page
2 (8)

Contents

1 Overview	3
1.1 Significant Prior Art	3
1.2 Eclipse Arrowhead architecture philophosy	4
2 Eclipse Arrowhead engineering process	6
3 Implementation specifics	7
4 References	7
5 Revision History	8
5.1 Amendments	8
5.2 Quality Assurance	8



1 Overview

This document describes the fundational principles for Eclipse Arrowhead architecture and implementation platform.

1.1 Significant Prior Art

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

A set of EU projects have built the foundation for what's 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 architecture 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 contained for its intended operation enabling local security, protection and if equipped with TDMA network MAC real time properties can be achieved.
 - System are self contained for its intended operation, e.g. owning and responsible for its own data storage and computational resources.
 - Arrowhead:
 - * Objective to be interoperability to legacy and internet protocols and being Open Source.
 - * Basic SOA foundation established, Look-up, Late binding and Loosely coupled.
 - * Mandatory core systems defined: ServiceRegistry, Orchestration, Authorisation
 - * Interoperability enabled through translation dynamically instantiated when needed.
 - * v3.3 released as open source
 - Productive4.0:
 - * Arrowhead Framework becomes Eclipse Arrowhead architecture and implementation platform
 - * Extending the implementation platform - the Arrowhead technology stack is defined
 - * v4.5 released
 - Arrowhead Tools
 - * Objective to reduce engineering cost with 20-50%
 - * Extending the Eclipse Arrowhead technology stack
 - * v4.6 released
 - * Achieved 30-95% engineering cost and time reduction in 28 industrial use cases along the extended IEC 81346 engineering process.

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 available at www.github.com/eclipsearrowhead.

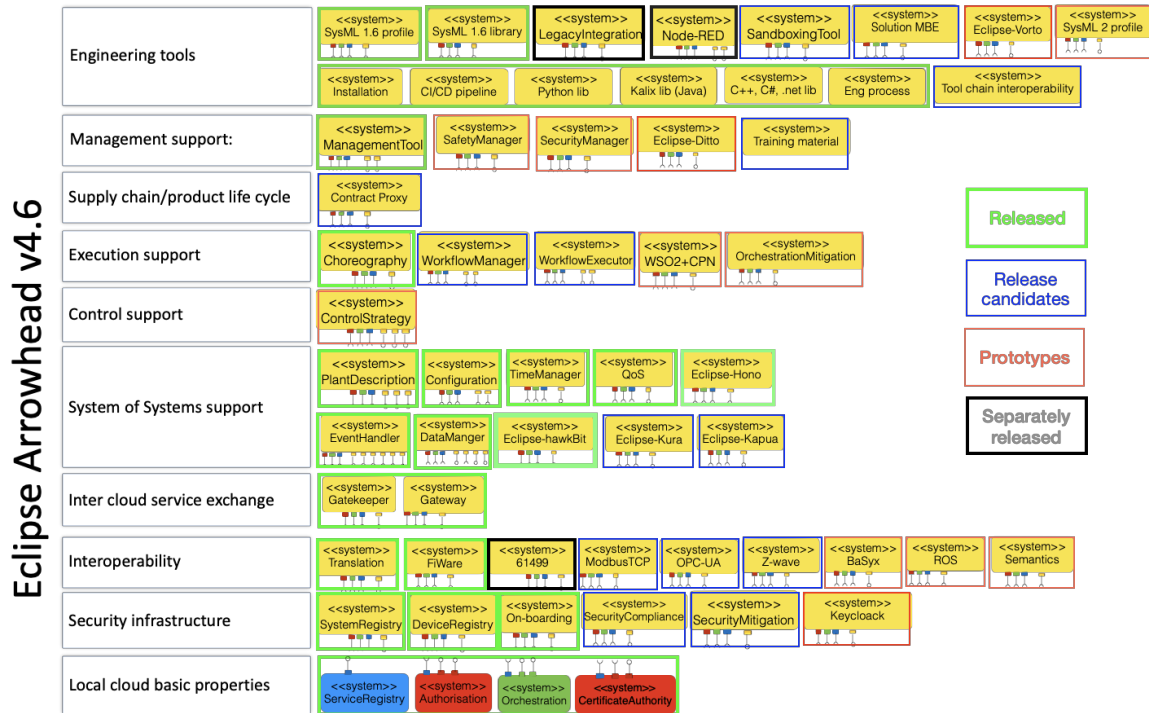


Figure 1: The Eclipse Arrowhead technology stack an dassociated microsystems, released, release candidates and prototypes.

1.2 Eclipse Arrowhead architecture philoposy

The architecture philiposy is based on the following key technology decision 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 enabling 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 storage 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.
 - Architectural levels defined are depicted in Figure 3.
- Technology objectives:

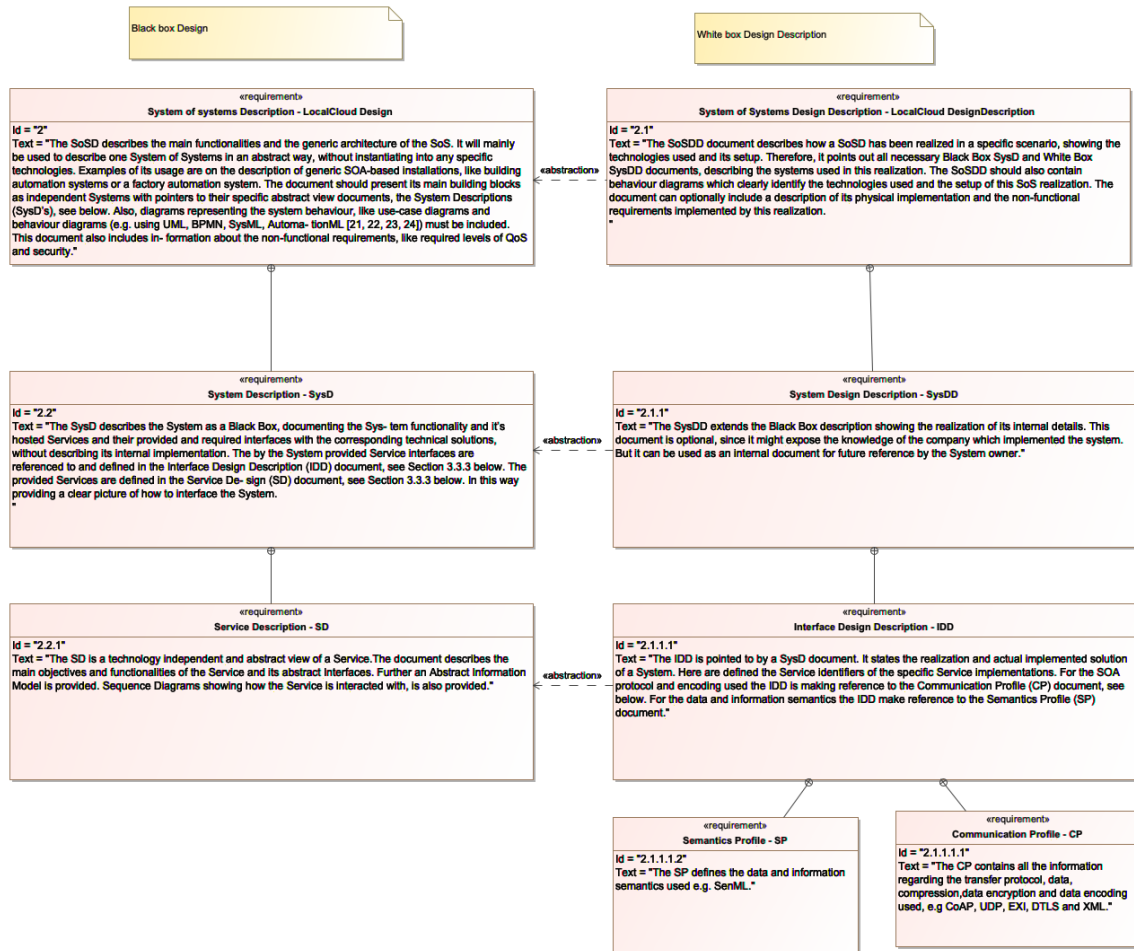


Figure 2: The Eclipse Arrowhead documentation structure.

- Interoperability support at Service level shall be provided regarding: SOA, IP and legacy protocols, encodings, compressions, security, data models, through translators or dedicated adaptors. For data model interoperability 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 than service level is being addressed. Security is highly recommended but can be discarded if desired.
- Secure on-boarding: On-boarding based on authentication of devices, microsystems and microservices shall be supported.
- Support for multiple strategy direction. A strategy direction may be: Security, LifeCycle, Maintenance, Business, Audit, Monitoring, BusinessAdministration, BusinessModels. This will also require ways of addressing interdependencies between the various strategy 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].

A set of terminologies used in the Eclipse Arrowhead developments and supporting project are important to highlight. The following are important to highlight 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:

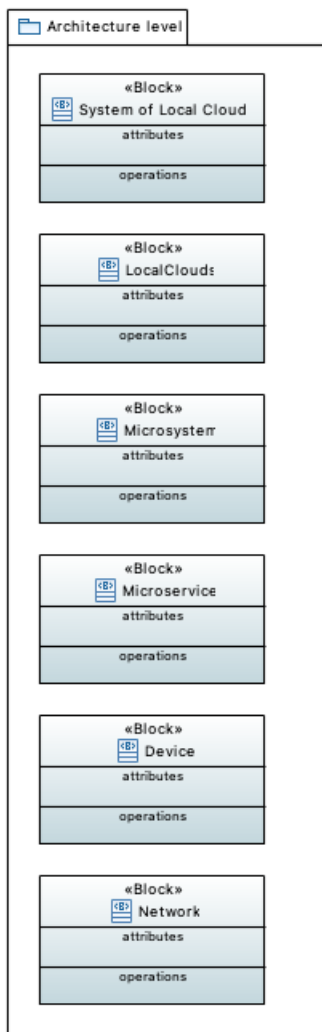


Figure 3: The Eclipse Arrowhead architectural levels from network to System of Systems.

- Service -> Microservice
- System -> Microsystem

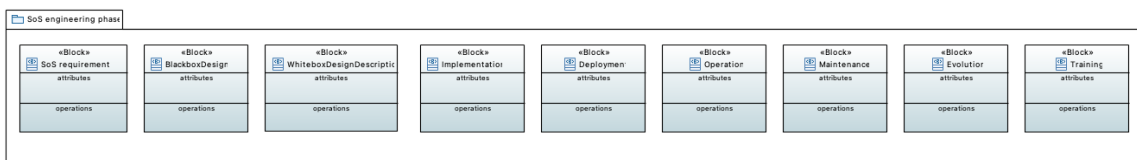


Figure 4: The Eclipse Arrowhead engineering process is an extension of the IEC81346 engineering process..

2 Eclipse Arrowhead engineering process

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



3 Implementation specifics

For the current implementations a set of technologies has been chosen for various reasons ranging from simple to use over “this are the technologies I’m familiar to” to there are the technologies “which provides what I want”. Technologies used for implementations of Eclipse Arrowhead microsystem are:

- Protocols: HTTP (REST), CoAP, MQTT, Websocket
- Network protocols: Ethernet, WiFi, 802.15.4
- Datamodel standards used: SenML
- Security protocol: X.509 certificates and tokens, see [8]
- Programming language: Java, Python
- Libraries: Java, C
- High performance: GO lang

4 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-oriented system-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] 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.



ARROWHEAD

Document title
Fundational principles
Date
2023-05-28

Version
4.6.1
Status
DRAFT
Page
8 (8)

5 Revision History

5.1 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				

5.2 Quality Assurance

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