

Technology comparison between ORiN2 and OPC UA

Ver. 1.0
May 11, 2015



ORiN Consortium Technical Committee

Introduction

- This document is a technical comparison based on the original implementation of OPC Client & Server by ORiN Consortium, and is not a document approved by related organizations, companies, or individuals.
- This document focuses on the comparison of “technologies” and does not include the comparison of “organizations”.
- Don’t know all the specifications of OPC UA correctly, so there may be some mistakes. If you point them out, we will review and correct them as soon as possible.

References :

1. OPC Unified Architecture specification (IEC 62541)
<https://jp.opcfoundation.org/developer-tools/specifications-unified-architecture>
2. OPC Unified Architecture (ISBN 978-3-540-68898-3)
Authors: Mahnke, Wolfgang, Leitner, Stefan-Helmut, Damm, Matthias
3. Softing OPC UA C++ Server and Client Development Toolkits for Windows

Overview

	OPC UA	ORiN 2
Support groups	OPC Consortium	ORiN Consortium
Formal name	<u>OPC</u> <u>U</u> nified <u>A</u> rchitecture	<u>O</u> pen <u>R</u> esource interface for the <u>N</u> etwork
Logo		
URL	www.opcfoundation.org	www.orin.jp
Birthplace	US	Japan
Time of birth	2008	2005
Secretariat (Japan)	Microsoft	JARA
Old Specifications	OPC-Classic (OPC-DA, A&E, HAD)	ORiN 1
Standard	IEC62541	ISO20242-Part4

Support range

	OPC UA	ORiN 2
Platform (Product-based)	Windows,Linux,Mac,Android	Windows,Linux,Mac,Android
Development Language	C,C++,C#,.NET,Java	C,C++,C#,.NET,Java,LabView,OLE(Delphi,Python,MATLAB,Ruby,...)
Other standards	OPC-Classic (OPC-DA, A&E, HAD)	ORiN1, CORBA, OPC DA, OPC UA, MESX, DDE, UPnP, SNMP, DirectShow, DirectInput, FTP, HLA, ICMP, ISO16100, ISO20242, Modbus, POP3, SMTP, RAC, DeviceNet, S-LINK, S-LINKV, IPLink
Target fields	FA/HA	FA/HA/ROBOT

Supported Technologies

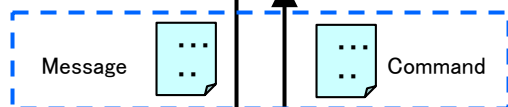
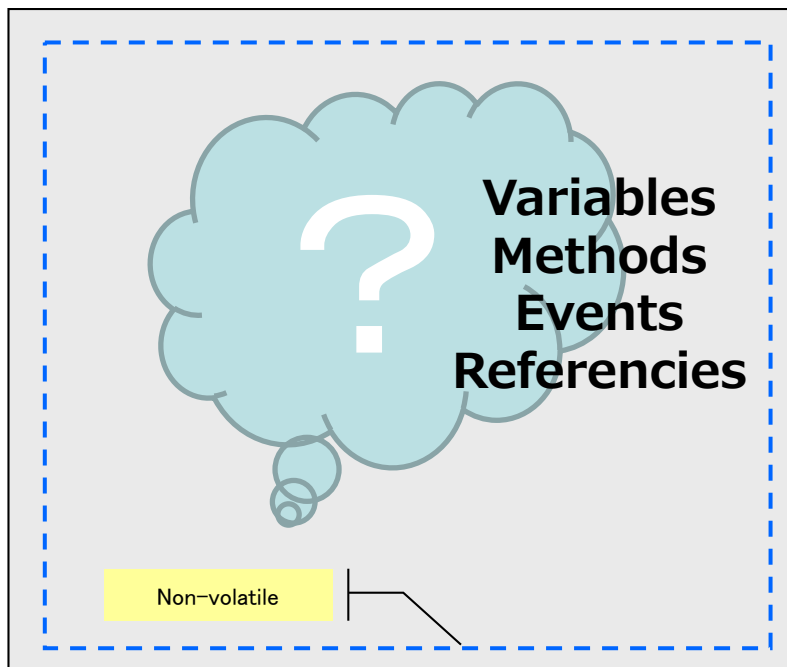
	OPC UA	ORiN 2
API	OPC-UA (Unified Architecture)	CAO (Controller Access Object)
Distributed Object Technology	Proprietary (?)	DCOM, CORBA, Java-RMI, b-CAP(proprietary)
WEB Service	SOAP/HTTP	[SOAP] CAP (Controller Access Protocol) [HTTP] e-CAP
P2P communication	UA TCP (UA Binary)	RAC, b-CAP (TCP/UDP)
Standard resource representation format	XML(?)	[XML] CRD (Controller Resource Definition)
Security	OpenSSL	[OpenSSL] B-CAP secure
Discovery service (Server search)	Dedicated servers are supported.	None

Resources Model

Target resource

OPC-UA

Controller (ex. Robot, PLC, NC)

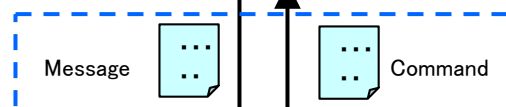
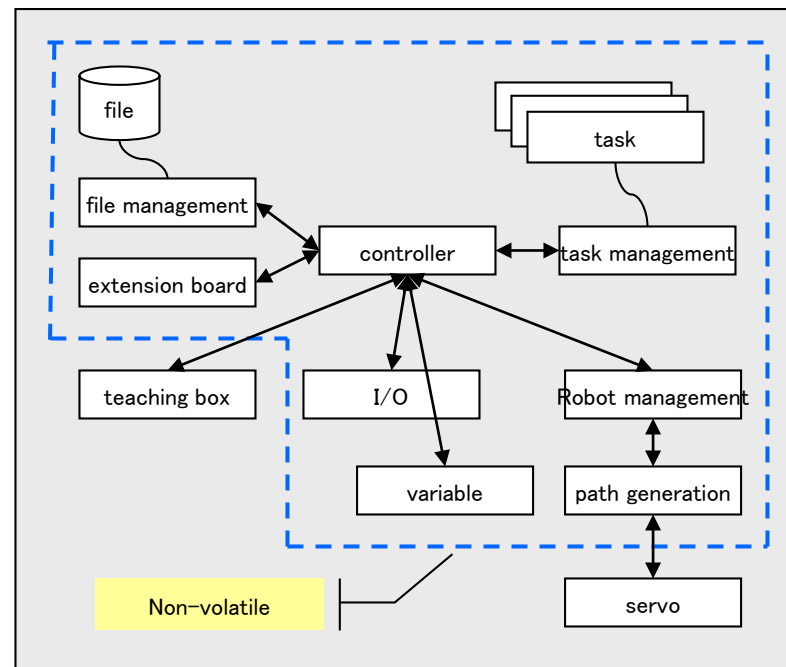


volatile

Subject is abstract and not limited

ORiN 2

Controller (ex. Robot, PLC, NC)



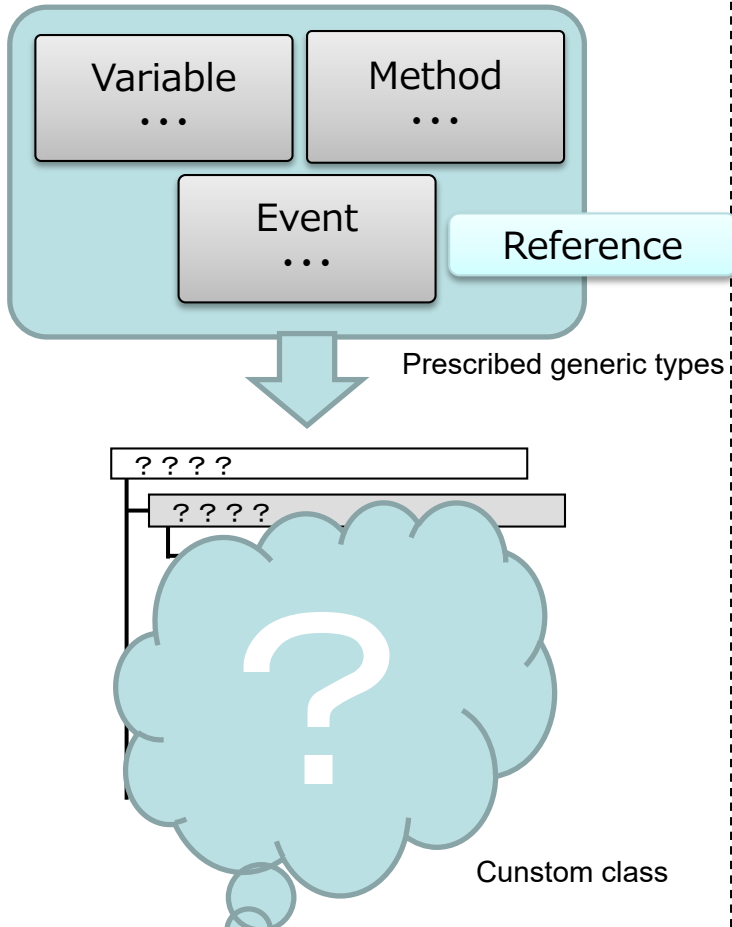
volatile

Target is assumed to be specific

Object Model

OPC UA

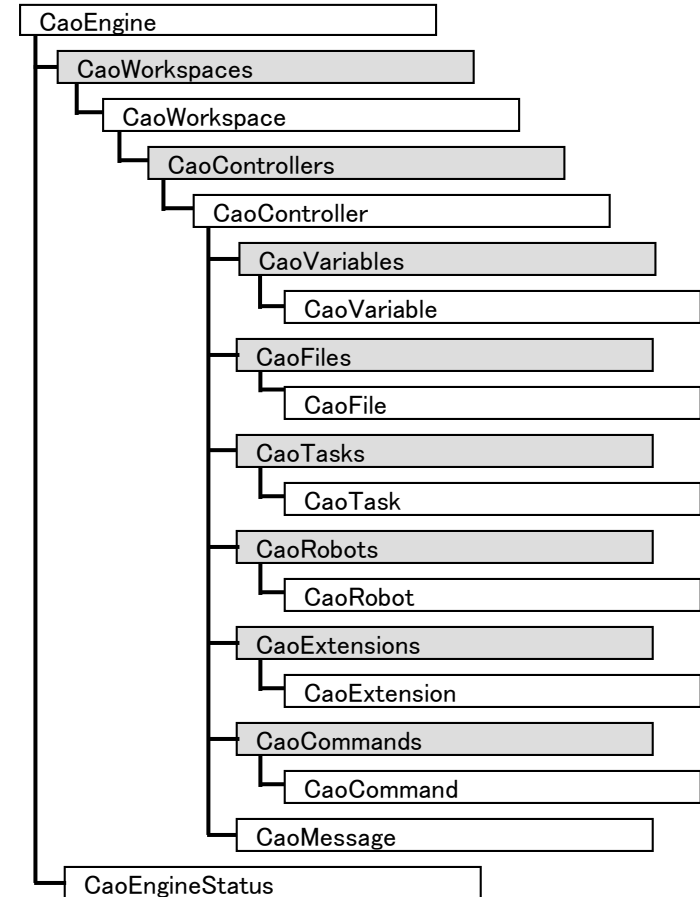
■ UA Automation Interface



Not specified, free to classify

ORiN(CAO engine)

■ CAO engine (dual interface)



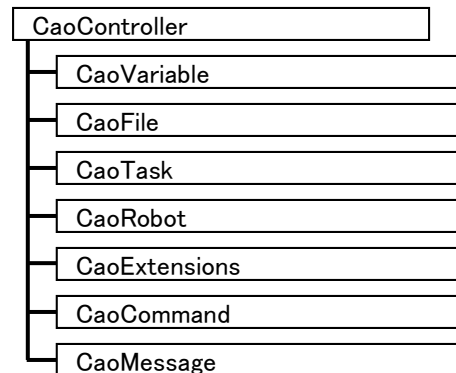
Prescribed classes will be provided.

Object Model (continued)

OPC UA

ORiN(CAO provider)

■ CAO provider (dual interface)



Note: OLE standard interface is not included

Number of classes: 19 (8)

Number of custom interfaces: 19 (8)

Total number of methods: 207 (109)

Figures in parentheses are CAO provider interfaces

Not specified, free to classify

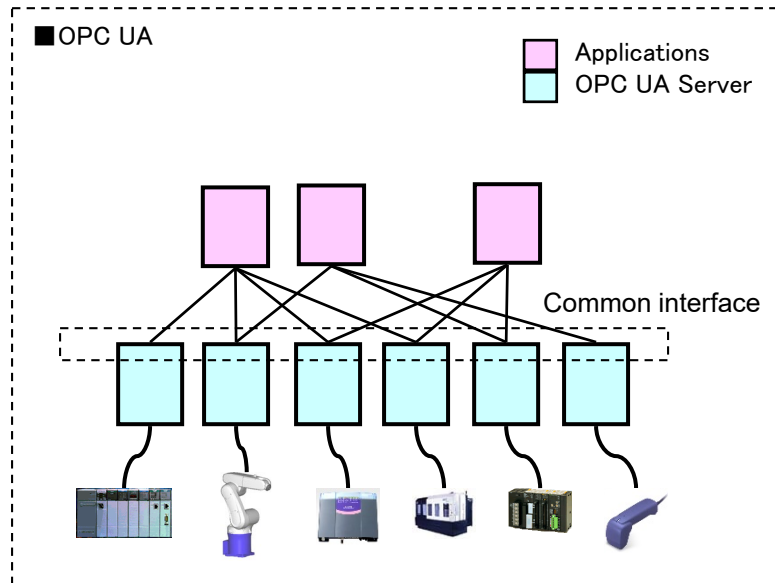
Prescribed classes will be provided

Considerations1 : Object Model

- Since the components of OPC UA objects (Variables, Methods, Events, Referencies) can be freely defined, and objects can be hierarchically, parallelized, and related in complex ways, complex systems can be modeled as they are.
- The model of ORiN (CAO) is specified, and the application side can be accessed by different devices in the same way. In addition, the provider side contains only the essential model, which is much to implement than OPC UA.

Architecture

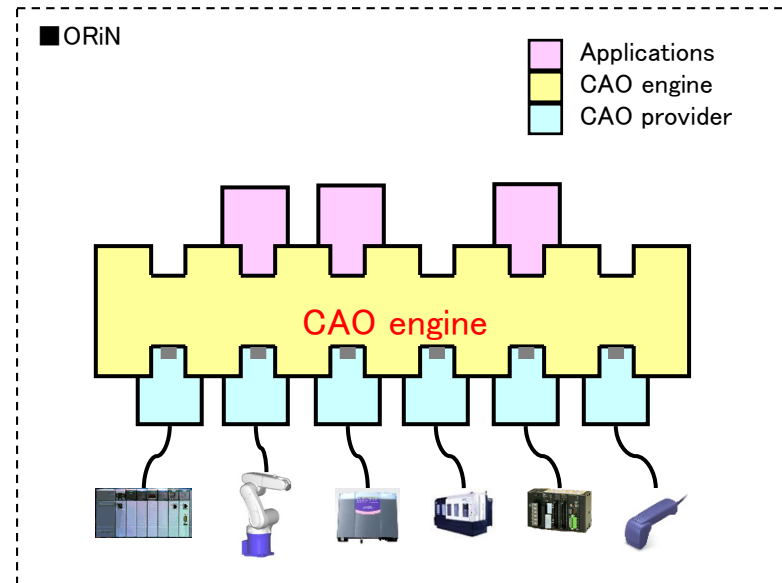
OPC UA



■ Features:

- Each application **operates the OPC UA server**
- Each device manufacturer implements its own OPC UA server

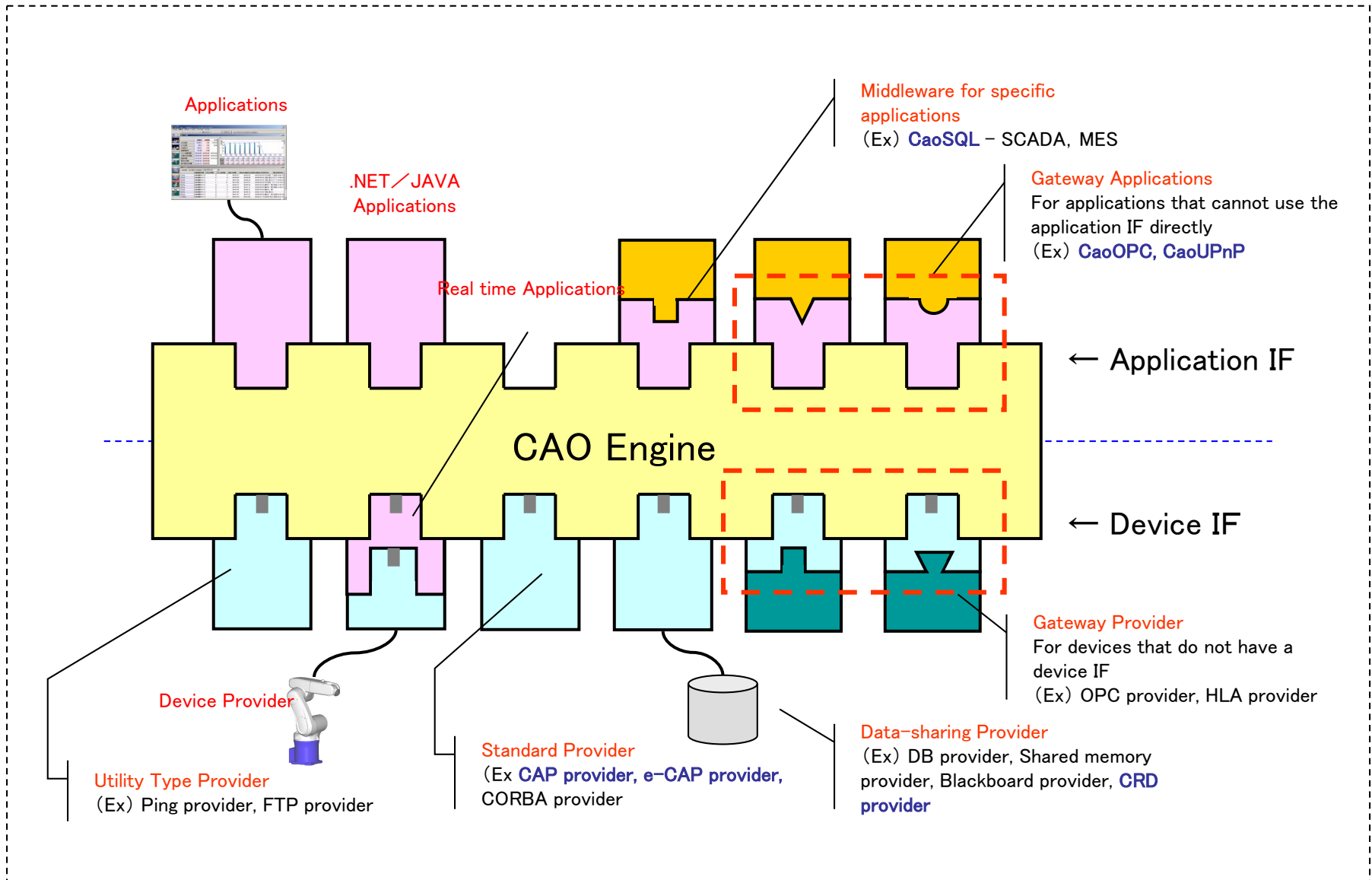
ORiN



■ Features:

- Each application **manipulates the abstracted device through the interface** provider by the engine
- **Reduce** the difference between the abstracted device and the real device **by the provider**
- The engine provides common functionality
- Each device manufacturer implements a provider

(Reference) Architecture of CAO



(Reference) Architecture

- OPC UA is a two-tier model in which the application directly controls the server. Therefore, the server needs to support the OPC UA standard.
- ORiN(CAO) is a three-layer model, where the application operates the device abstracted by CAO, and the provider reduces the difference from the real device. Therefore, even if the actual device is too old to support the new standard, the device can be used by the provider.

Summary

- ORiN and OPC UA are different specifications developed with different backgrounds.
- While OPC UA prioritizes freedom and uses a sophisticated object model, ORiN prioritizes connectivity and uses a well-defined object hierarchy model to cover a wide range of scalability from lower-layer embedded devices to higher-layer enterprise applications.
- Therefore, the objectification approach (OPC UA=concretization, ORiN=abstraction) is different, and the architectural design policy (OPC UA=two-tier, ORiN=three-tier) is also different.
- Therefore, it can be said that OPC UA is best suited for connecting complex upper layer systems and ORiN is best suited for connecting a wide variety of lower layer devices, and both are complementary to each other.