UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Implementation Object Linking and Embedding for Processes Control Unified Architecture Specification on Secure Device

The future standard for communication and information modeling in automation

Yuankui Wang

Agenda

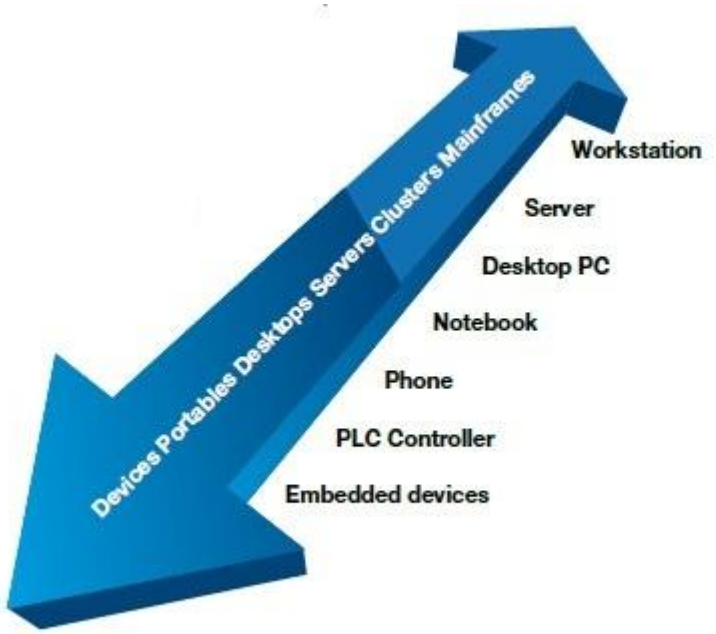
- Introduction and Motivation
- OPC Unified Architecture Specification
- Smart Card Technology
- Implementation Scenario
- Goals
- Time Lines
- Reference

Introduction and Motivation

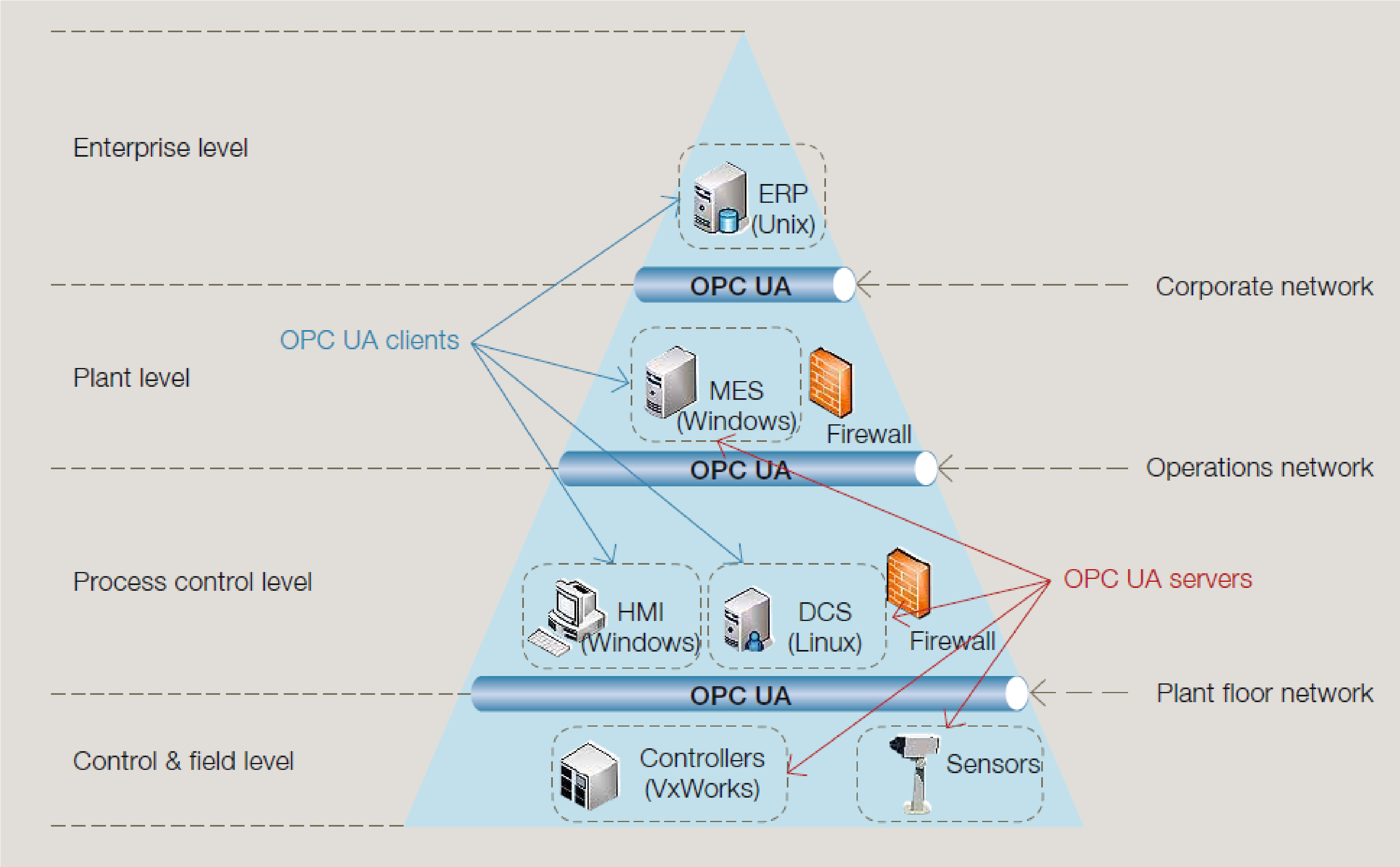
- In industry automation world, Machine-to-Machine technology is widely applied.
- Exchange gather information during collaborative machining process
- motion control in legacy networks
- Over 22,000 products supplied by over 3,200 vendors
- Crucial: system interconnectivity, common interface for communication, security
- Classic OPC offers solutions for data access, historical data access, alarms and events.
- But there exits limitations and imperfections
- Windows platform only, DCOM/COM, no complex data structure

OPC Unified Architecture Specification

- Platform independent data communication
- Standardized communication via internet and firewalls
- Protection against unauthorized access
- Availability and reliability
- SOA architecture
- Object oriented meta model
- Simplification by unification

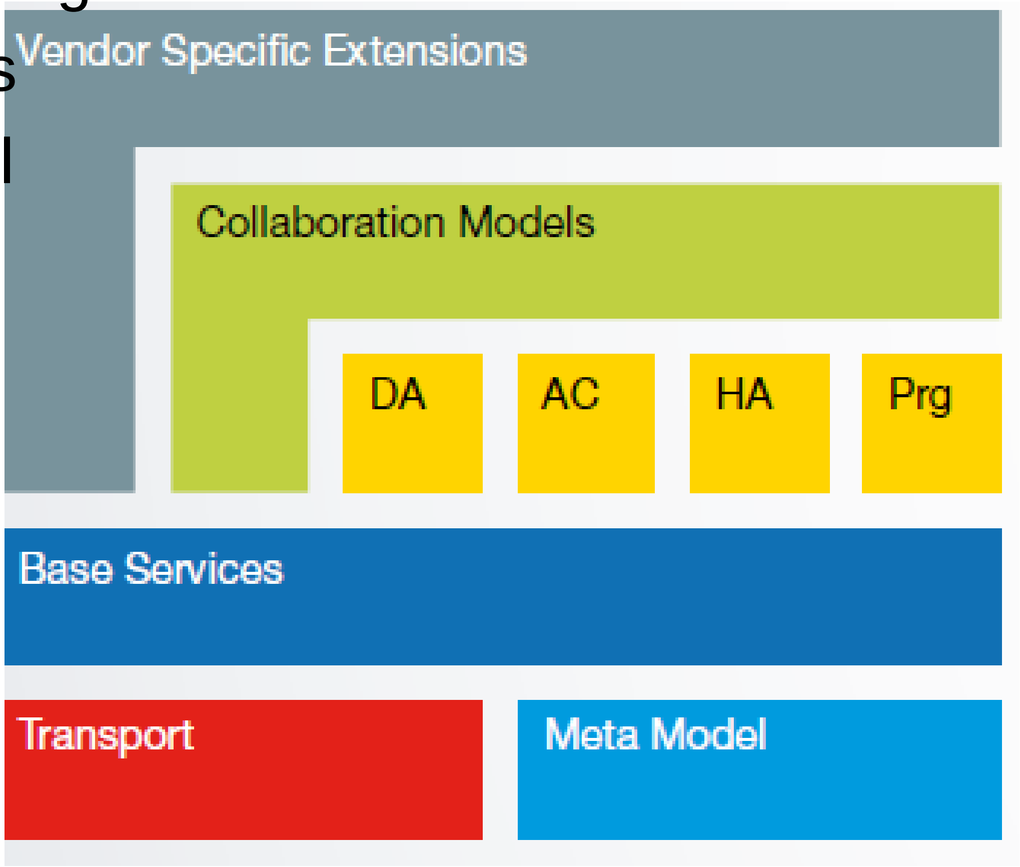


OPC UA Specification



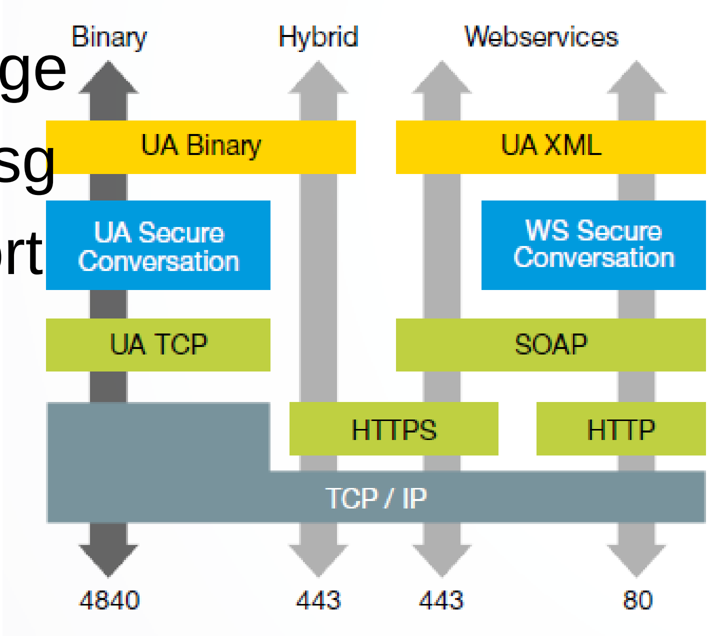
Key components of Unified Architecture

- Object oriented modeling capabilities
- Transport protocol bindings
- Fix set of base services
- OPC information model
- Extendable



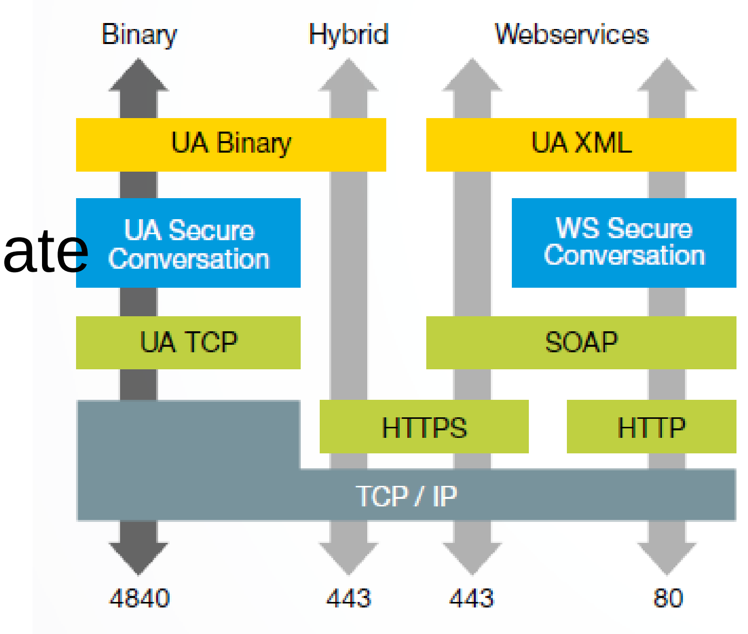
Protocol Binding

- Native UA Binary(mandatory)
  - Extremely fast and optimized
  - Preferred protocol between embedded devices
- HTTPS with UA Binary
  - Implemented low end,midrange
  - UA binary content in Https Msg
  - Using TLS encrypted transport security



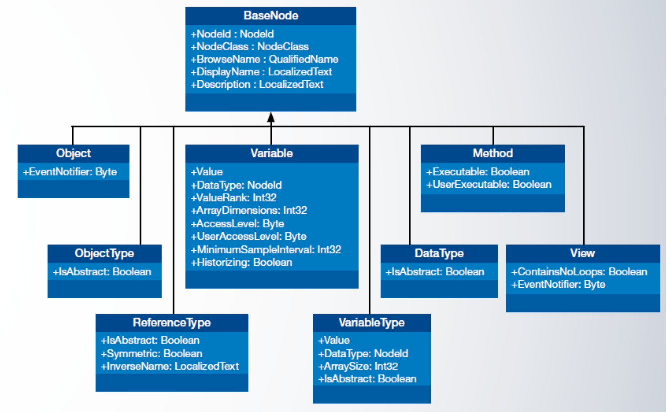
Protocol Binding

- HTTPS with SOAP and XML Encoding
  - Hybrid for web client application
  - Or in cases only port 443 can be used
- HTTP with SOAP and WS secure Conversation and XML Encoding
  - High level system
  - Only permitted to communicate on port 80



Data Model

- OPC UA defines a generic object model including the corresponding type system
- Generic date model
- Modeling rules how physical sys can be transformed in an UA conformant model
- Based on Data Model, information model is developed
  - Enhance the basic set of model
  - Data access
  - Alarms
  - Conditions
  - Historical Access
  - Programs



BaseNode
+NodeId : NodeId
+NodeClass : NodeClass
+BrowseName : QualifiedName
+DisplayName : LocalizedText
+Description : LocalizedText

Object
+EventNotifier: Byte

Variable
+Value
+DataType: NodeId
+ValueRank: Int32
+ArrayDimensions: Int32
+AccessLevel: Byte
+UserAccessLevel: Byte
+MinimumSampleInterval: Int32
+Historizing: Boolean

Method
+Executable: Boolean
+UserExecutable: Boolean

ObjectType
+IsAbstract: Boolean

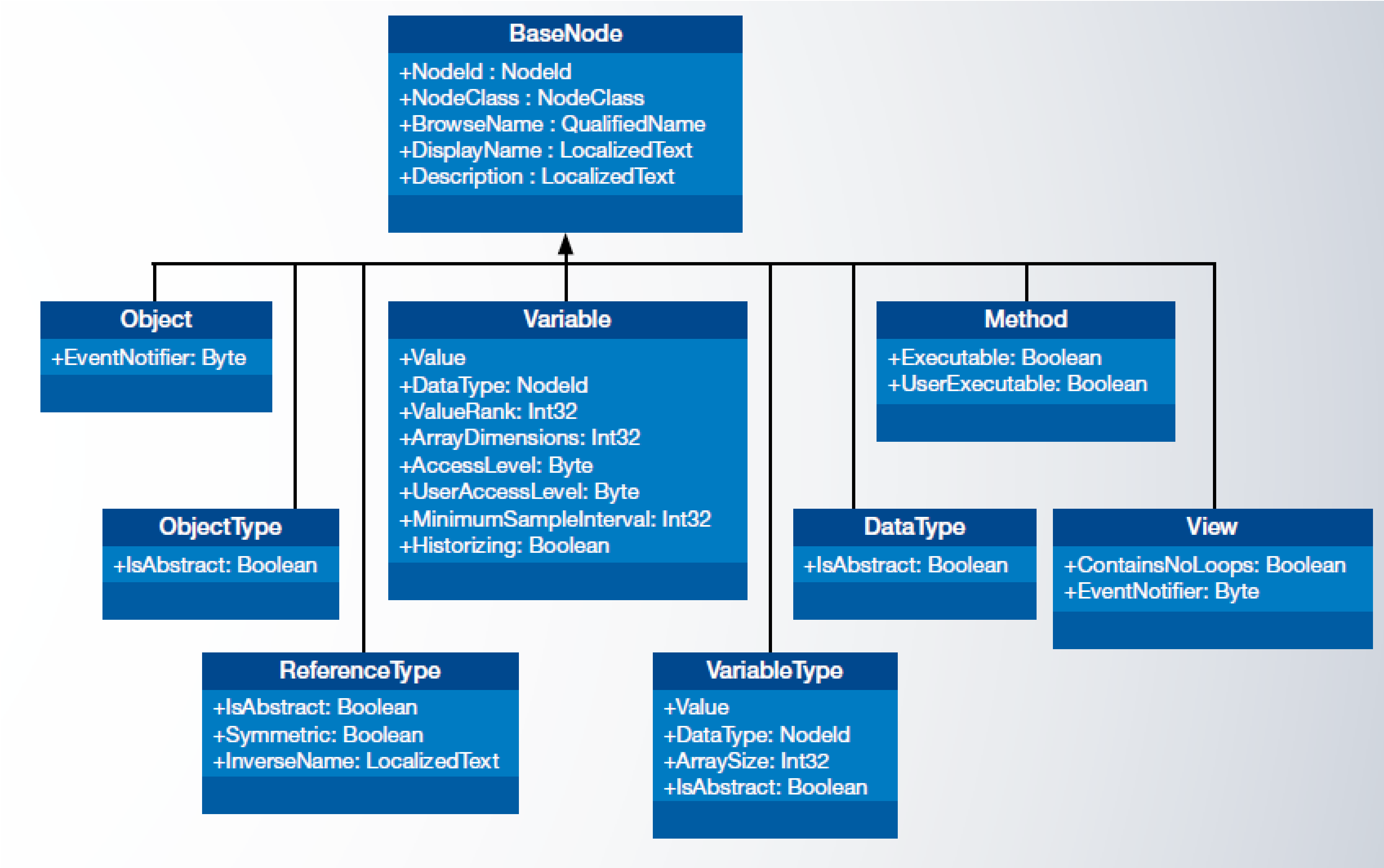
DataType
+IsAbstract: Boolean

View
+ContainsNoLoops: Boolean
+EventNotifier: Byte

ReferenceType
+IsAbstract: Boolean
+Symmetric: Boolean
+InverseName: LocalizedText

VariableType
+Value
+DataType: NodeId
+ArraySize: Int32
+IsAbstract: Boolean

Data Model



BaseNode
+NodeId : NodeId
+NodeClass : NodeClass
+BrowseName : QualifiedName
+DisplayName : LocalizedText
+Description : LocalizedText

Object
+EventNotifier: Byte

Variable
+Value
+DataType: NodeId
+ValueRank: Int32
+ArrayDimensions: Int32
+AccessLevel: Byte
+UserAccessLevel: Byte
+MinimumSampleInterval: Int32
+Historizing: Boolean

Method
+Executable: Boolean
+UserExecutable: Boolean

ObjectType
+IsAbstract: Boolean

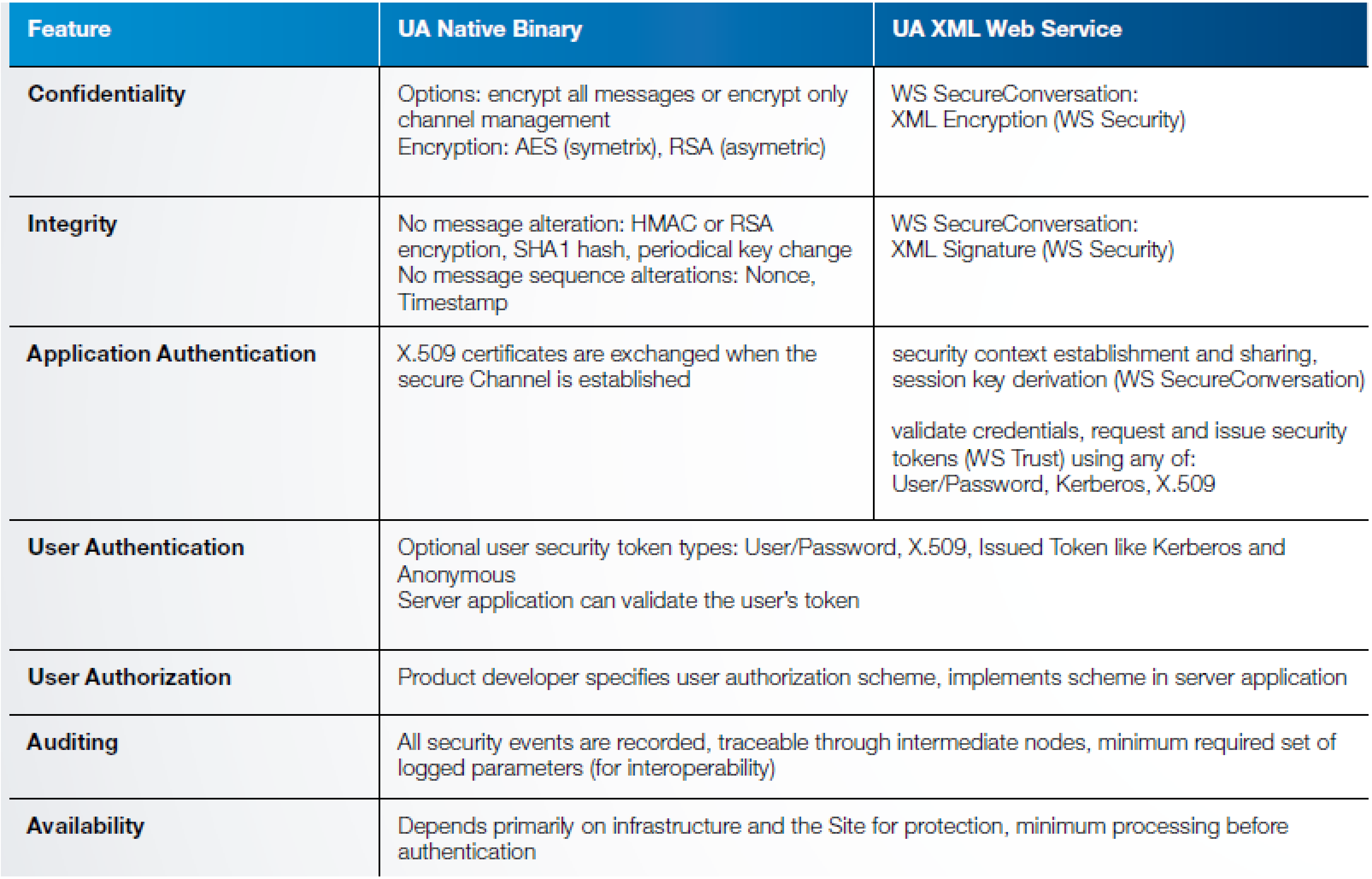
DataType
+IsAbstract: Boolean

View
+ContainsNoLoops: Boolean
+EventNotifier: Byte

ReferenceType
+IsAbstract: Boolean
+Symmetric: Boolean
+InverseName: LocalizedText

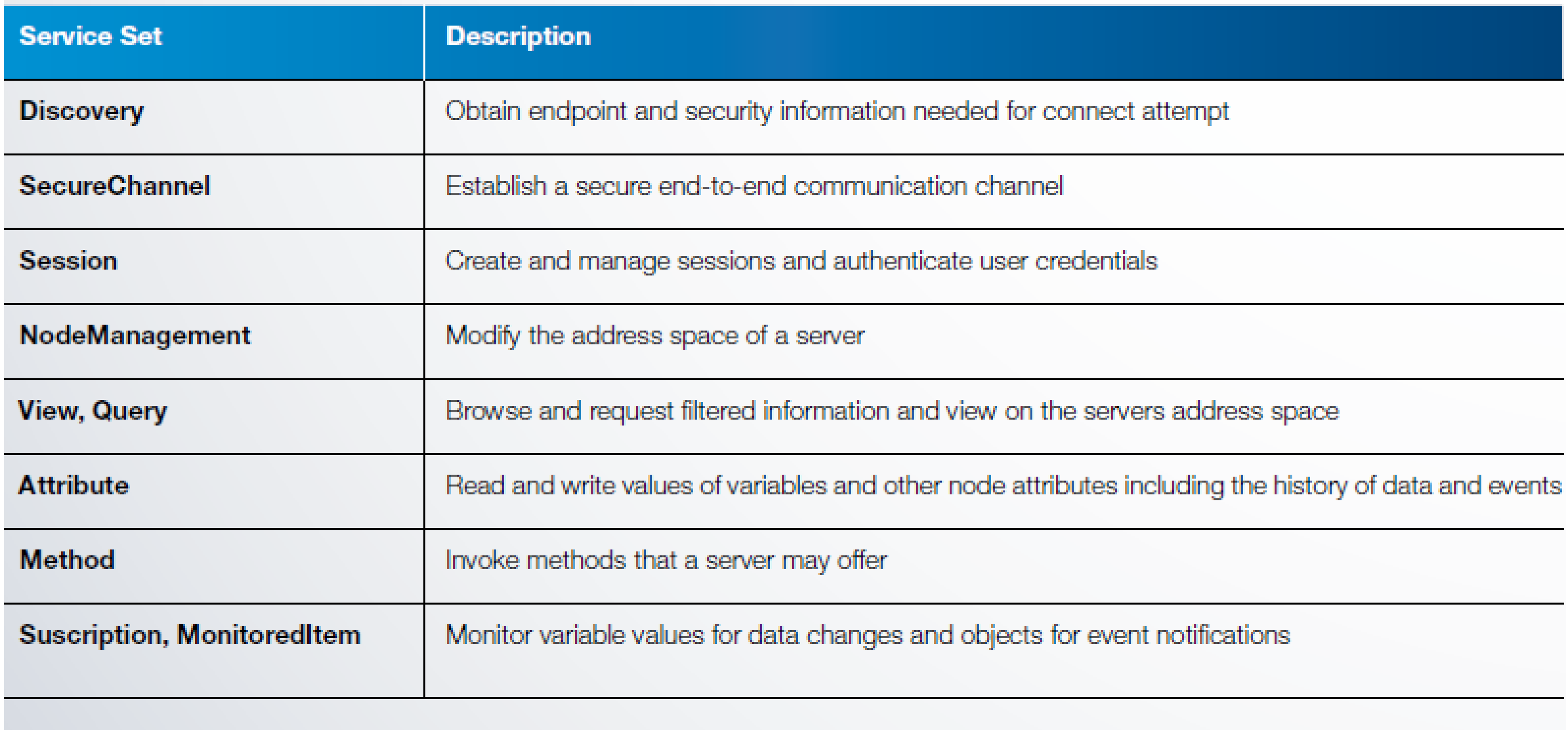
VariableType
+Value
+DataType: NodeId
+ArraySize: Int32
+IsAbstract: Boolean

Security



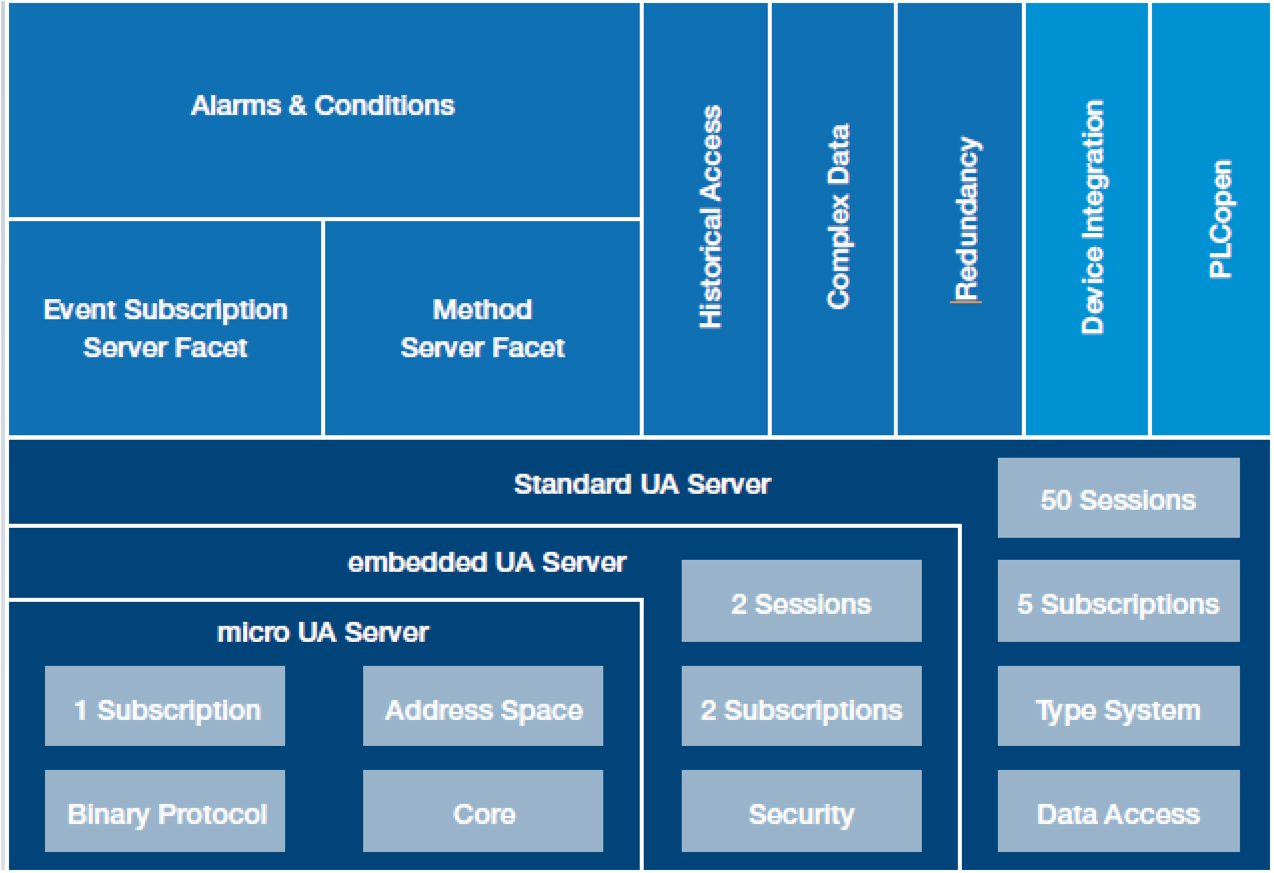
Feature	UA Native Binary	UA XML Web Service
Confidentiality	Options: encrypt all messages or encrypt only channel management Encryption: AES (symetrix), RSA (asymetric)	WS SecureConversation: XML Encryption (WS Security)
Integrity	No message alteration: HMAC or RSA encryption, SHA1 hash, periodical key change No message sequence alterations: Nonce, Timestamp	WS SecureConversation: XML Signature (WS Security)
Application Authentication	X.509 certificates are exchanged when the secure Channel is established	security context establishment and sharing, session key derivation (WS SecureConversation) validate credentials, request and issue security tokens (WS Trust) using any of: User/Password, Kerberos, X.509
User Authentication	Optional user security token types: User/Password, X.509, Issued Token like Kerberos and Anonymous Server application can validate the user's token	
User Authorization	Product developer specifies user authorization scheme, implements scheme in server application	
Auditing	All security events are recorded, traceable through intermediate nodes, minimum required set of logged parameters (for interoperability)	
Availability	Depends primarily on infrastructure and the Site for protection, minimum processing before authentication	

Services



Service Set	Description
Discovery	Obtain endpoint and security information needed for connect attempt
SecureChannel	Establish a secure end-to-end communication channel
Session	Create and manage sessions and authenticate user credentials
NodeManagement	Modify the address space of a server
View, Query	Browse and request filtered information and view on the servers address space
Attribute	Read and write values of variables and other node attributes including the history of data and events
Method	Invoke methods that a server may offer
Suscription, MonitoredItem	Monitor variable values for data changes and objects for event notifications

OPC UA Architecture



Alarms & Conditions

Event Subscription Server Facet

Method Server Facet

Historical Access

Complex Data

Redundancy

Device Integration

PLCopen

Standard UA Server

50 Sessions

embedded UA Server

2 Sessions

5 Subscriptions

micro UA Server

1 Subscription

Address Space

2 Subscriptions

Type System

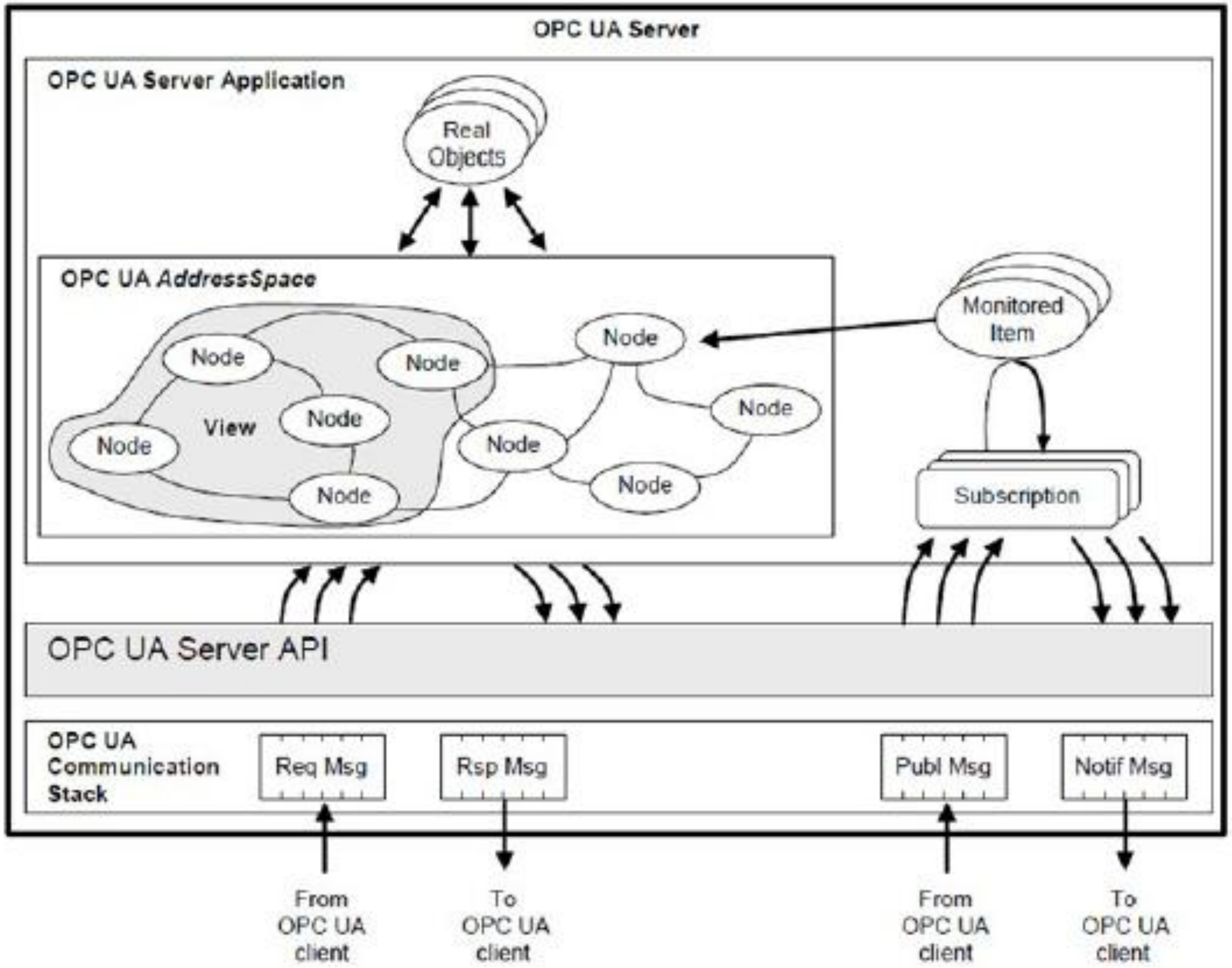
Binary Protocol

Core

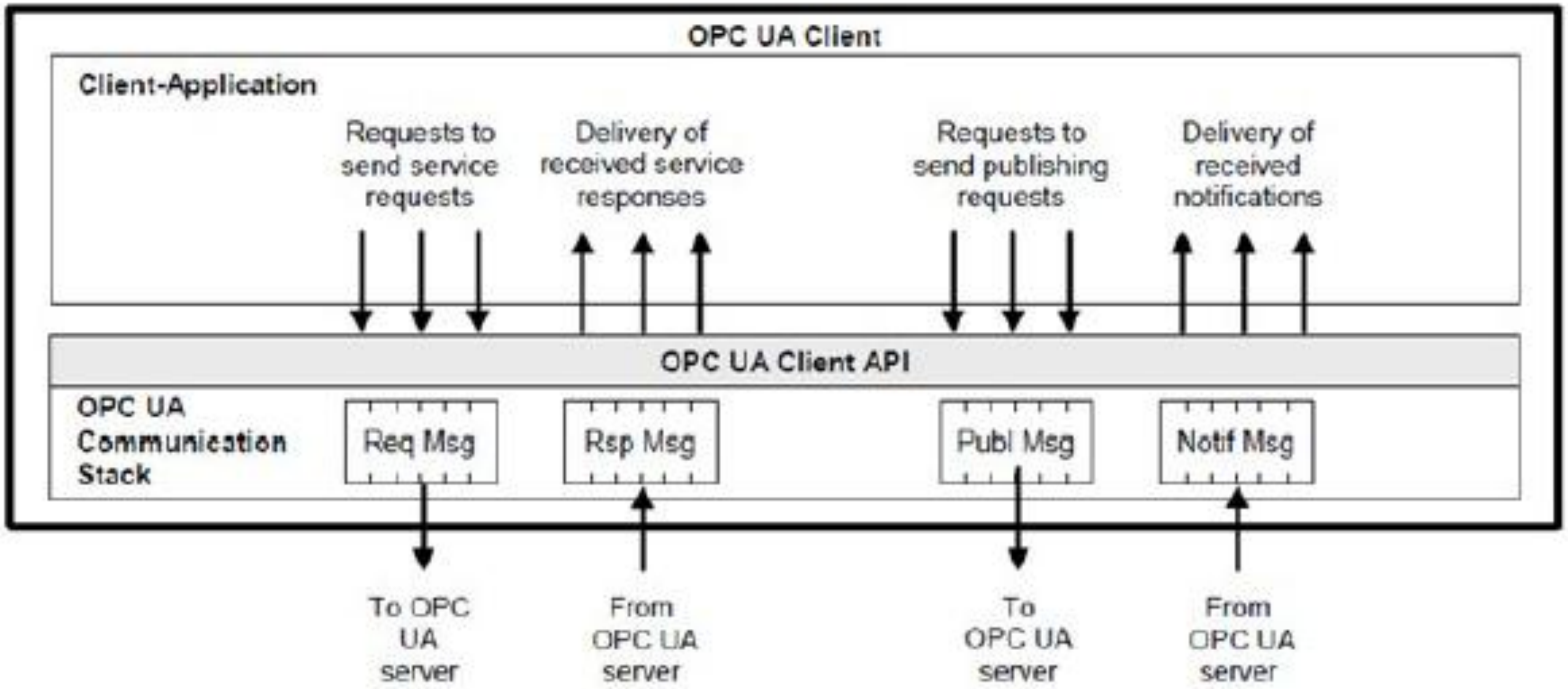
Security

Data Access

OPC UA Architecture



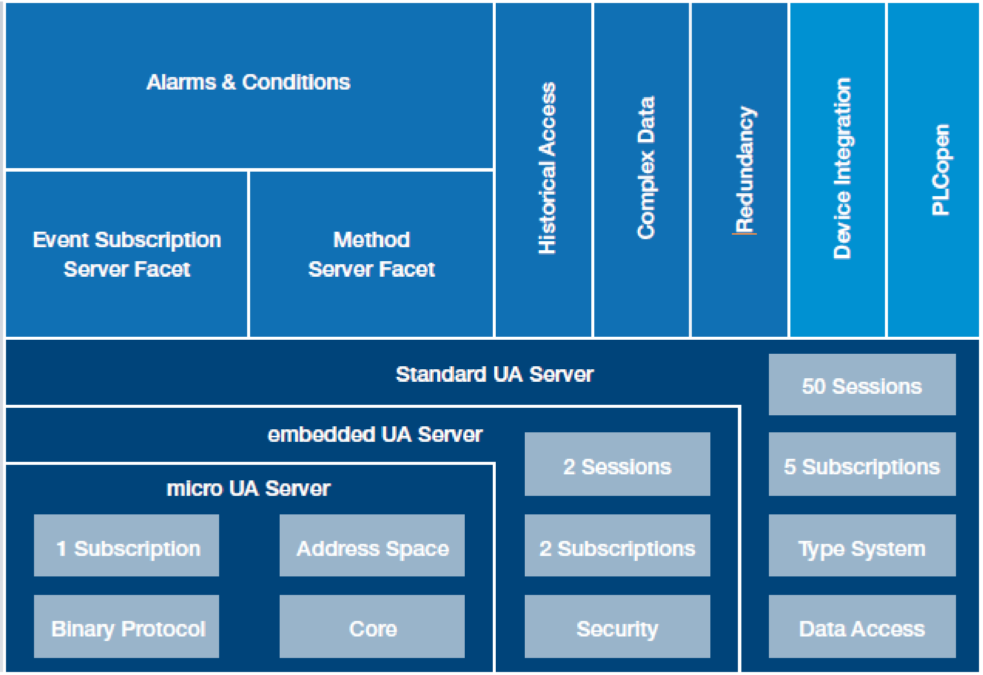
OPC UA Architecture


OPC UA Client
Client-Application
Requests to send service requests
Delivery of received service responses
Requests to send publishing requests
Delivery of received notifications
OPC UA Client API
OPC UA Communication Stack
Req Msg
Rsp Msg
Publ Msg
Notif Msg
To OPC UA server
From OPC UA server
To OPC UA server
From OPC UA server

Smart Card Technology

- Finance, Communication, personal identification, payment
- APDU based communication between card and CAD
- Security token
- Process cryptographic algorithms on hardware
- Self-containment structure

OPC UA on Secure Device



Alarms & Conditions

Event Subscription Server Facet

Method Server Facet

Historical Access

Complex Data

Redundancy

Device Integration

PLCopen

Standard UA Server

50 Sessions

embedded UA Server

2 Sessions

5 Subscriptions

micro UA Server

1 Subscription

Address Space

2 Subscriptions

Type System

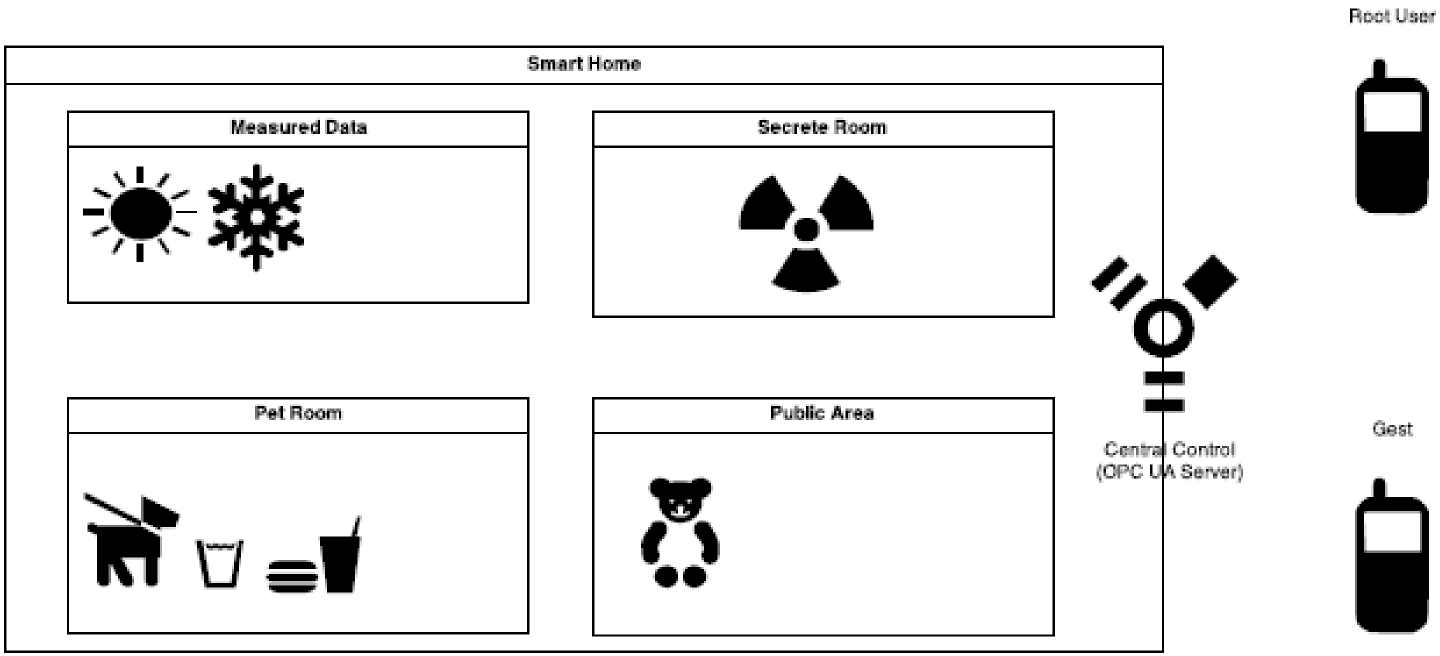
Binary Protocol

Core

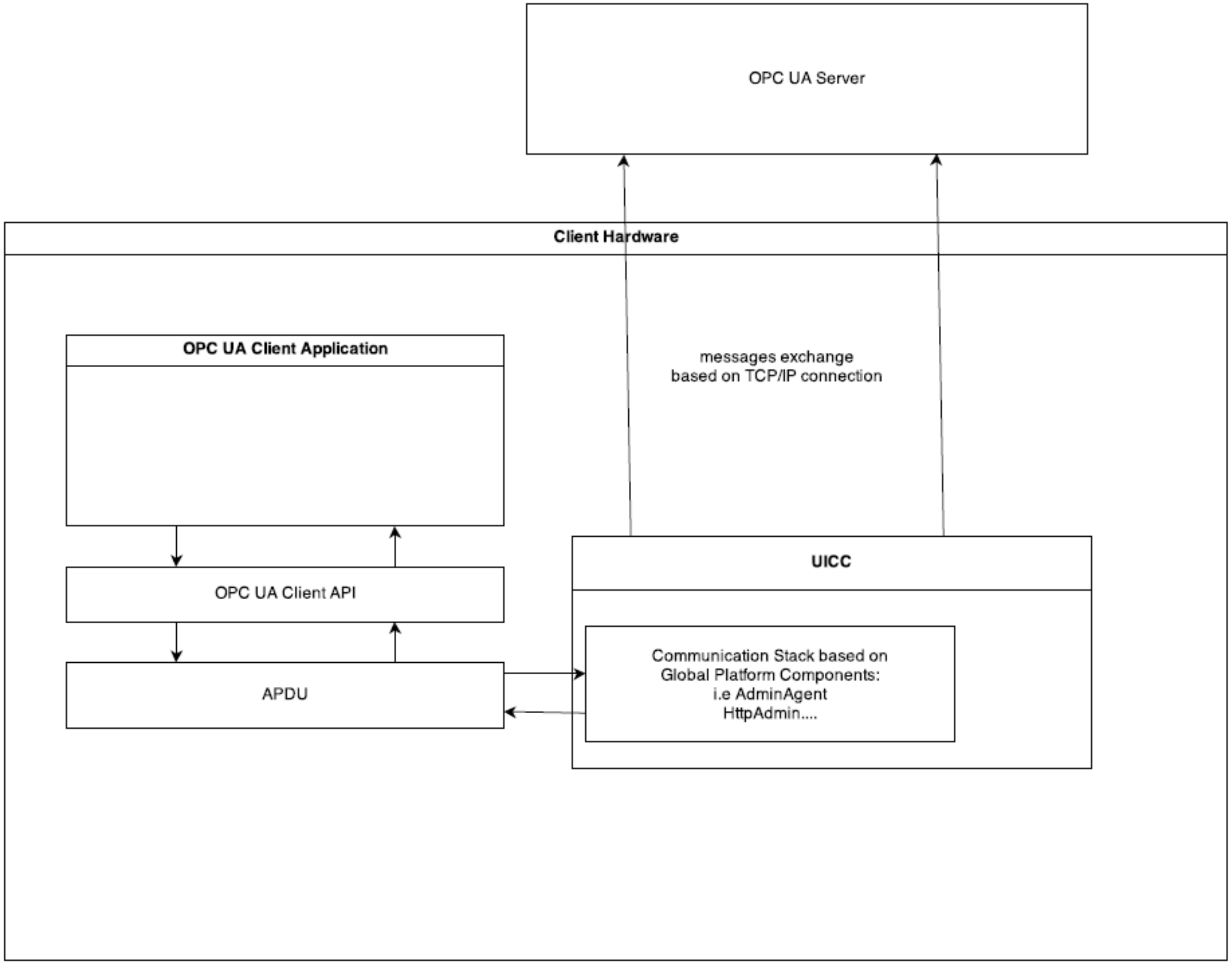
Security

Data Access

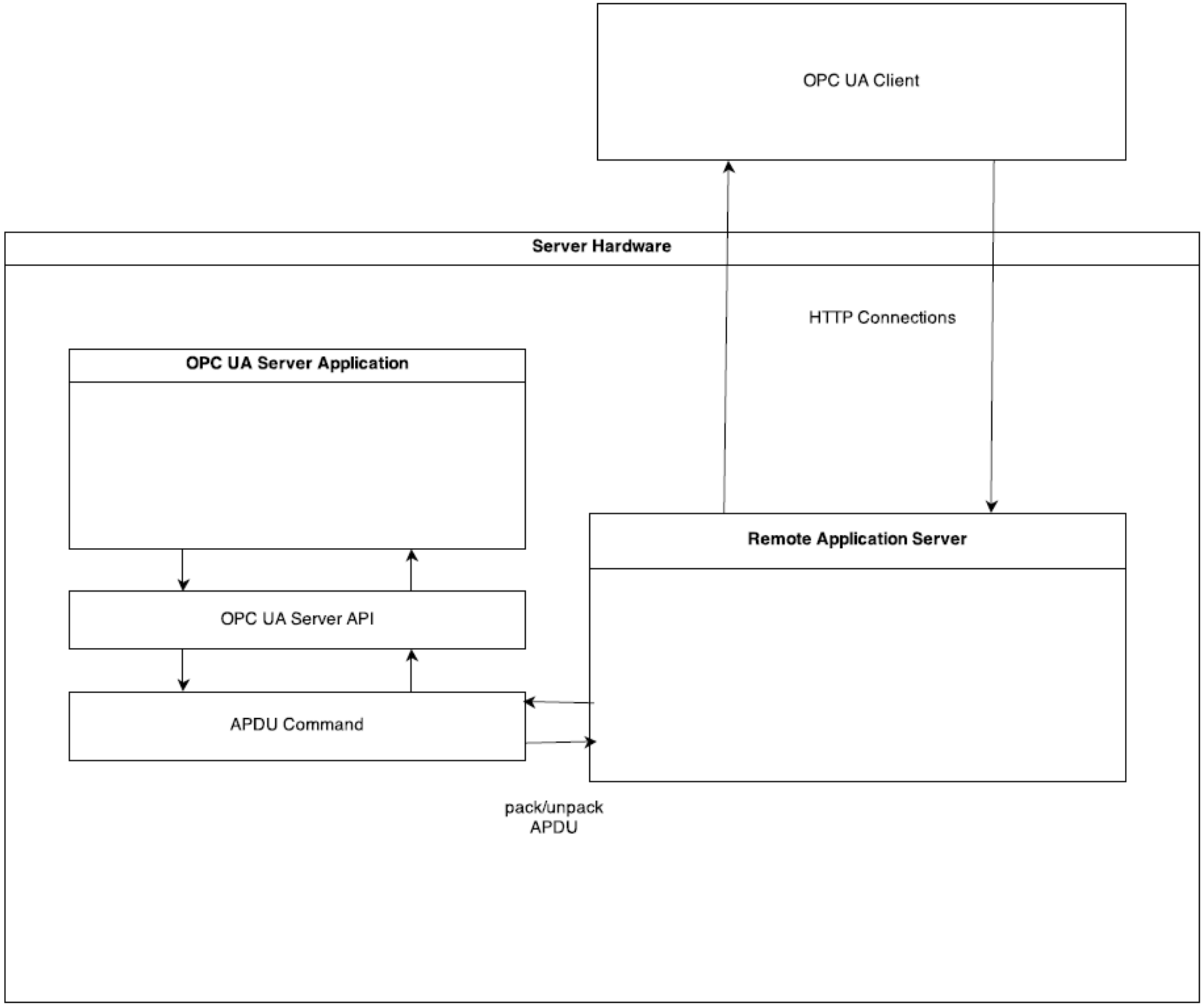
Implementation scenario



Implementation scenario


OPC UA Server
Client Hardware
OPC UA Client Application
messages exchange
based on TCP/IP connection
OPC UA Client API
UICC
APDU
Communication Stack based on
Global Platform Components:
i.e AdminAgent
HttpAdmin....

Implementation scenario



Goals

- Describing highlighting features of OPC UA
- Analyzing security protocols and their performance
- Studying smart card technology
- Learning security mechanisms provided by smart card
- Build OPC UA standard client and server application
- Build client side communication stack on smart card
- Build/simulate remote application server

Time Lines

- Reference
- Dummy client/server construction
- Communication stack on UICC smart card
- Remote application server construction
- Combination and debugging
- Analyze secure protocols
- Analyze performance

Rerferences

- OPC UA specification 1-11
- Stefan-Helmut Leitner and Wolfgang Mahnk: Opc ua-service-oriented architecture for industrial applications
- Wolfgang Mahnke, Stefan-Helmut Leitner:OPC Unified Architecture
- Wolfgang Rankl und Wolfgang Eng: Handbuch der chipkarten - 5. deutsche auflage. (2008)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Thank you! Question?