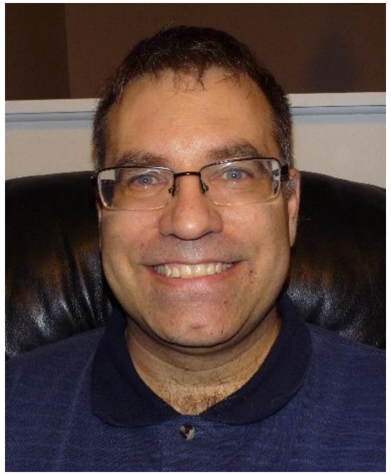
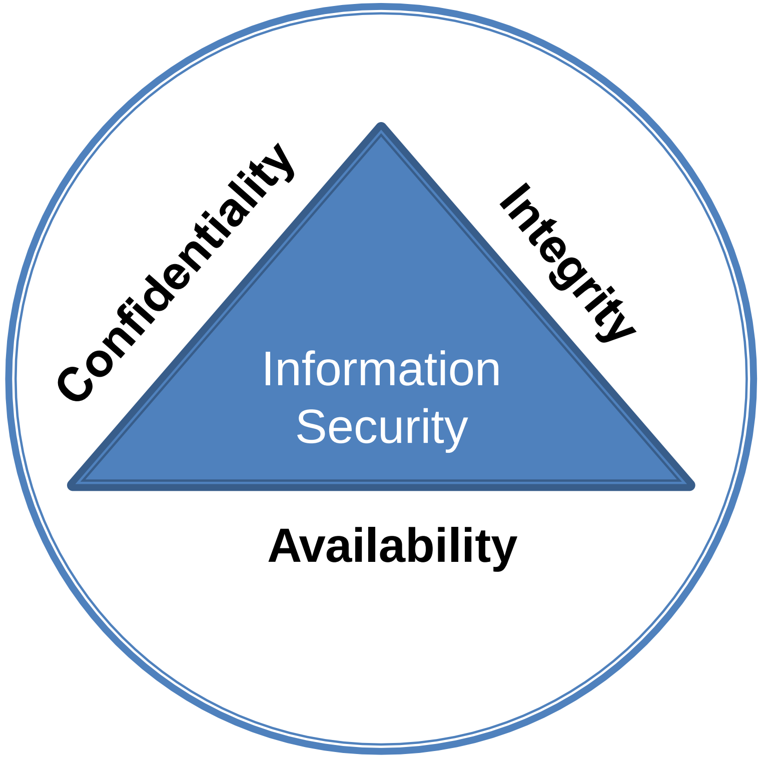
OPC UA Security Deep Dive



Randy Armstrong
Chair of OPC UA Security Working Group
OPC Foundation
randy.armstrong@opcfoundation.org

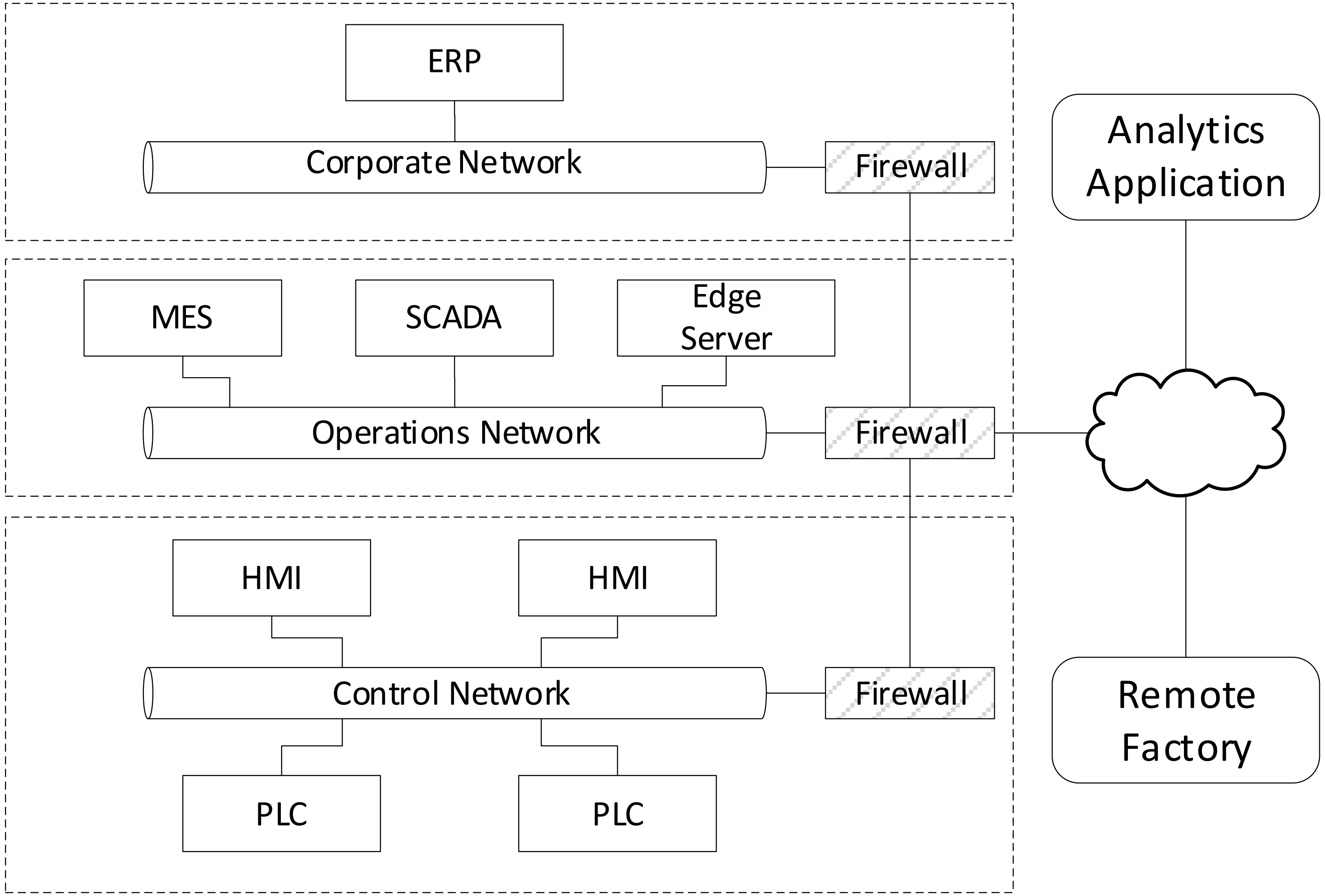
Key Security Concepts

- **Trusted Information (CIA triad)**
  - Confidentiality
  - Integrity
  - Availability
- **Access Control (AAA principle)**
  - Authentication
  - Authorization
  - Accounting (Auditability)
- **Configurability**
  - Providing Identities use for Authentication
  - Specifying Rules for Authorization



Security Environment

- Multiple Tiers and Multiple Networks
- Firewalls/NAT routers at multiple levels
- Multiple secure islands connected via the Internet.
- Local edge servers used to push information to cloud base analytics applications.



Security

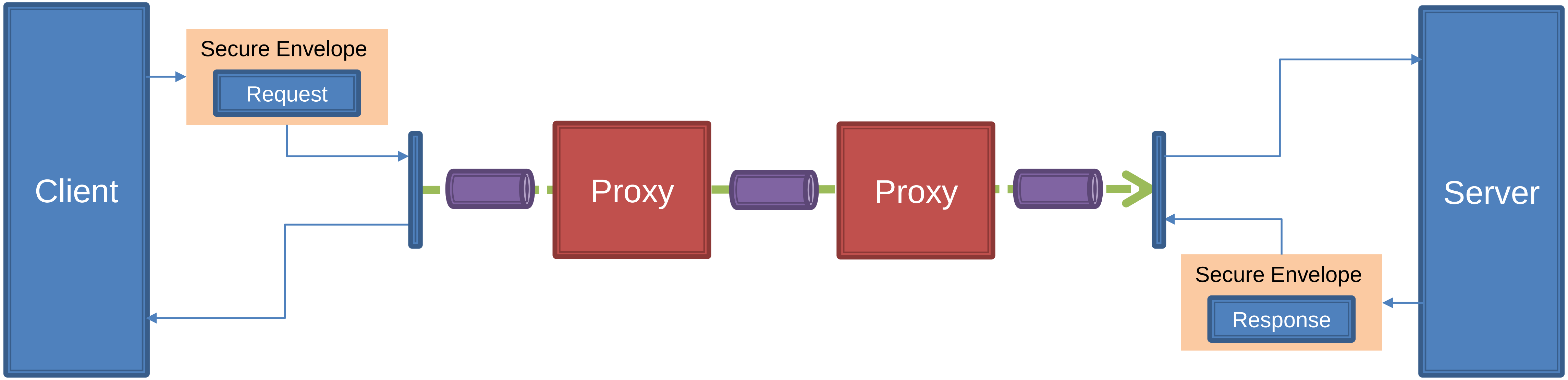
Security is incorporated into OPC UA at all levels!

- Confidentiality and integrity of communication is only part of the solution

- End-to-End Communication Security
- User Authentication
- Roles and role Management
- Audit logging
- Certificate management infrastructure

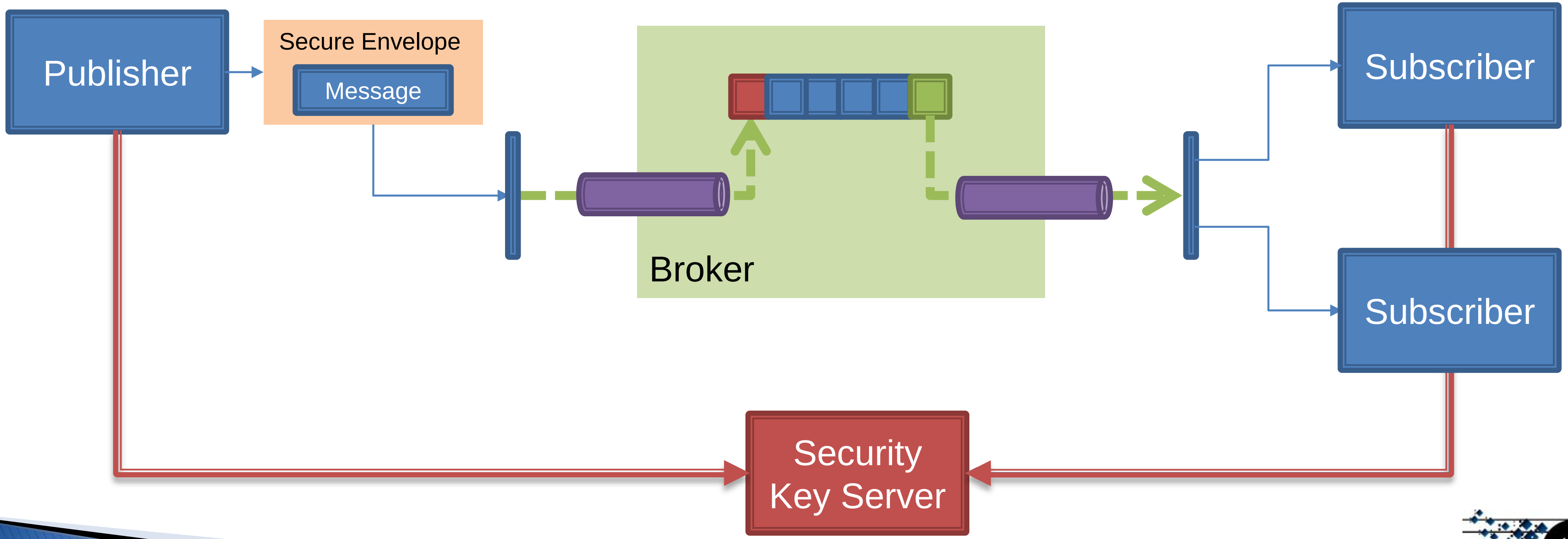
End to End Security in Client-Server

- Client-Server allows clients and servers to exchange messages;
- Messages are exchanged over a SecureChannel;
- A SecureChannel can be created over any transport
- A SecureChannel can be routed through untrusted proxies.



End to End Security in PubSub

- PubSub allows for publishers to send messages to multiple subscribers;
- Messages are sent via any transport;
- Messages are secured with keys supplied by Security Key Server;
- Keys are distributed out of band and can be fetched in batches;

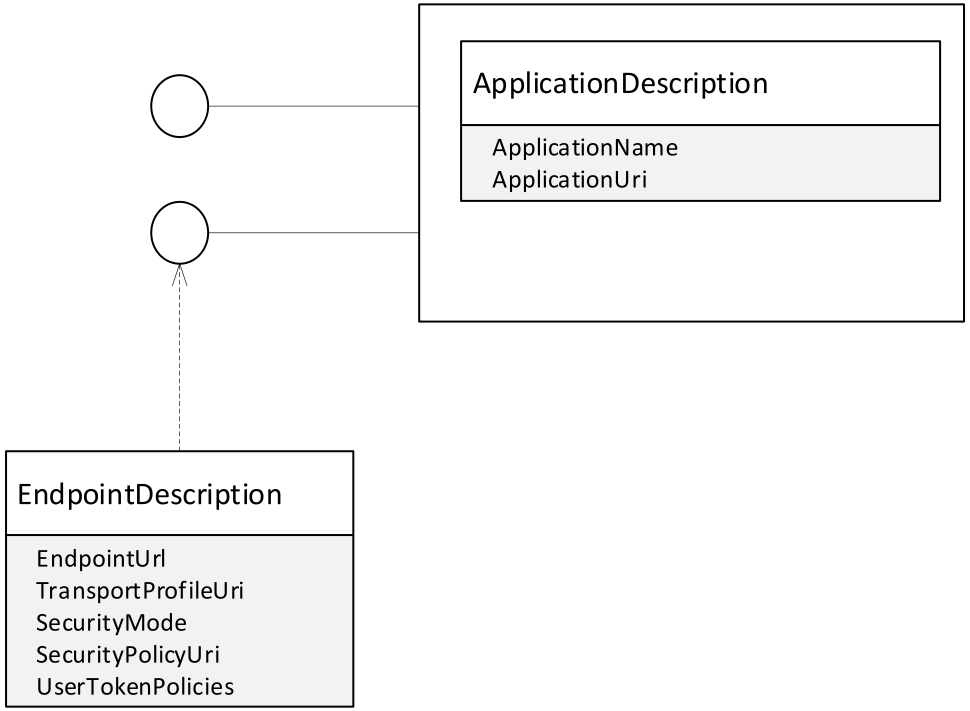


Applications and Endpoints

- Applications have a unique identifier.

  urn:hostname:company:product

- The ApplicationUri says WHO you are connecting to.
- Applications have multiple endpoints
- Endpoints may support multiple SecurityPolicies and/or SecurityModes.
- The EndpointDescription says HOW you connect.



Endpoint Descriptions

Field	Meaning
EndpointUrl	Not all Endpoints will be accessible from the Clients location. Servers with multiple NICs or behind NAT will expose additional Endpoints
SecurityPolicyUri	The SecurityPolicy to use. Client must choose the best policy that it supports.
SecurityMode	Sign or SignAndEncrypt Choose based on application requirements. Choose SignAndEncrypt when in doubt.
SecurityLevel	A number that allows Clients to determine which EndpointDescription is preferred by the Server. Clients should always select the highest unless they have a specific need to use something else.
DiscoveryUrls	URLs that can be used to fetch EndpointDescriptions. Connect with SecurityMode=None

Security Policies

- Suites of Algorithms identified by a URI:

http://opcfoundation.org/UA/SecurityPolicy#

  - None
  - Aes128_Sha256_RsaOaep [A]
  - Basic256Sha256 [B]
  - Aes256_Sha256_RsaPss
  - PubSub-Aes128-CTR [A]
  - PubSub-Aes256-CTR

- Enhances interoperability by reducing the number of permutations and combinations that need to be tested.

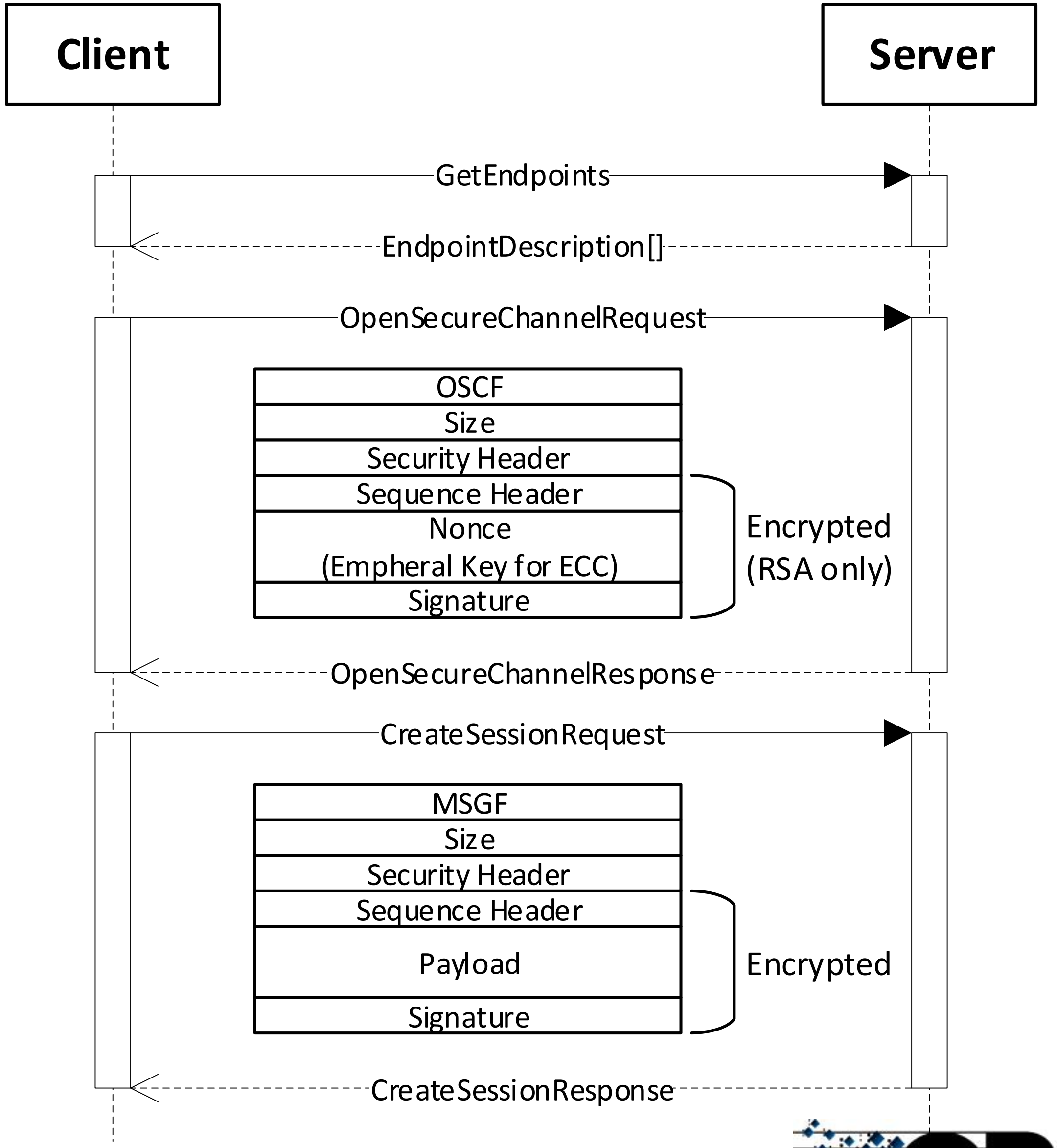
- SecurityPolicies supported by a Server are returned in the EndpointDescriptions.

Aes128_Sha256_RsaOaep
SymmetricSignatureAlgorithm_HMAC-SHA2-256
SymmetricEncryptionAlgorithm_AES128-CBC
AsymmetricSignatureAlgorithm_RSA-PKCS15-SHA2-256
AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA1
KeyDerivationAlgorithm_P-SHA2-256
CertificateSignatureAlgorithm_RSA-PKCS15-SHA2-256
Aes128-Sha256-RsaOaep_Limits

Initiating a Connection

- RSA based SecurityPolicies
  - Sign and Encrypt
  - Derive Keys
  - Use Symmetric Encryption

- ECC based SecurityPolicies
  - Sign + Ephemeral Keys
  - Derive Keys
  - Use Symmetric Encryption
  - Authenticated Encryption (AEAD) an option.



Client

Server

GetEndpoints

EndpointDescription[]

OpenSecureChannelRequest

OSCF

Size

Security Header

Sequence Header

Nonce

(Empheral Key for ECC)

Signature

Encrypted (RSA only)

OpenSecureChannelResponse

CreateSessionRequest

MSGF

Size

Security Header

Sequence Header

Payload

Signature

Encrypted

CreateSessionResponse

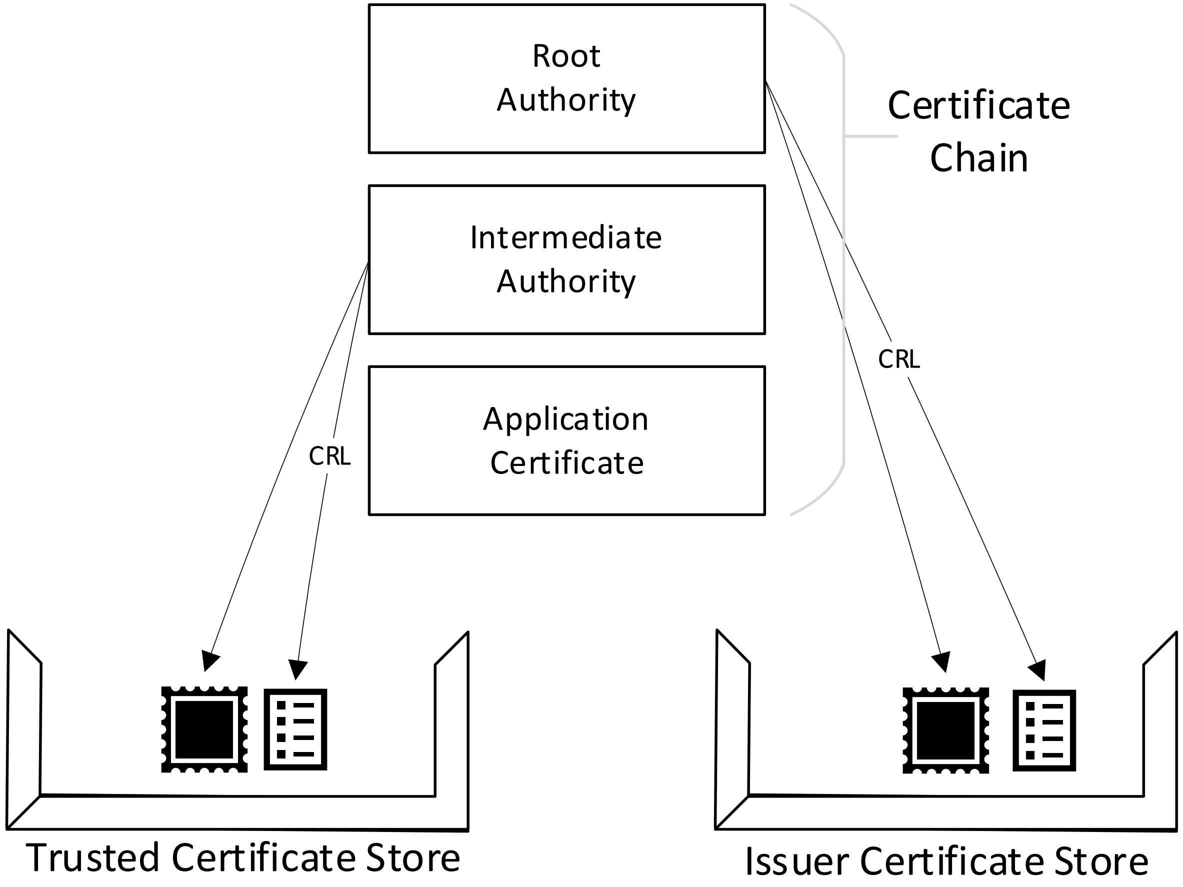
Application Authentication and Authorization

- Applications all have Certificates assigned
- Applications are uniquely identified by the ApplicationUri.
- The ApplicationUri is in all Certificates.
- DNS Name and/or IP Address an optional secondary identifier.
- Trust Lists are used to control access to Applications.

X509 Certificate Field	Description
SubjectName	CN=<Application Name>/O=<OwnerOperator Name>
SubjectAltName	URI:<Application Uri>;dnsName:<Host Name>;ipv4Address:<IP>

Determining Trust

- Leverages PKI infrastructure
  - Certificate Authorities issue Certificates
  - A Certificate and its issuers is a Chain
- Trust Lists have two stores:
  - Trusted Certificates
  - Issuer Certificates
- An Application is Authenticated if the Certificate and all issuers are valid and not revoked.
- An Application is Authorized (a.k.a. Trusted) if at least one Certificate in the chain is in the list of Trusted Certificates.
- Root authorities do not have to be in the Trust List.
- CRLs stored locally or remotely
- Chain may be transmitted on wire or preconfigured



Authenticating a Certificate

- Full rules found in Part 4:
  - https://reference.opcfoundation.org/v104/Core/docs/Part4/6.1.3/

- Errors are suppressible or non-suppressible.

- Suppressible errors:
  - Bad_CertificatePolicyCheckFailed
  - Bad_CertificateTimeInvalid
  - Bad_CertificateHostNameInvalid
  - Bad_CertificateUriInvalid
  - Bad_CertificateUseNotAllowed

- All suppressed errors must be logged!

- Clients only receive Bad_SecurityChecksFailed.
- Servers must provide a log that allows admins see the true error code.

Common Certificate Problems

Error	Action
Bad_CertificateUntrusted	Make the Certificate or one of its issuers is in the Trusted Certificates store
Bad_CertificateTimeInvalid	Check the system time on both machines to make sure it is current. Make sure a newly created Certificate has a valid from date in the past.
Bad_CertificateHostNameInvalid	Use a URL with correct hostname when connecting to the Server. Update Server configuration to use the correct hostname.
Bad_CertificateRevocationUnknown	Install a current CRL in the same store as the CA Certificate.
Bad_CertificateChainIncomplete	Configure application to send the complete chain or Install CA certificates in the peer’s issuer certificate store.
Bad_CertificatePolicyCheckFailed	Ensure Certificate key length is large enough. Check that the Certificate uses the correct Key type.

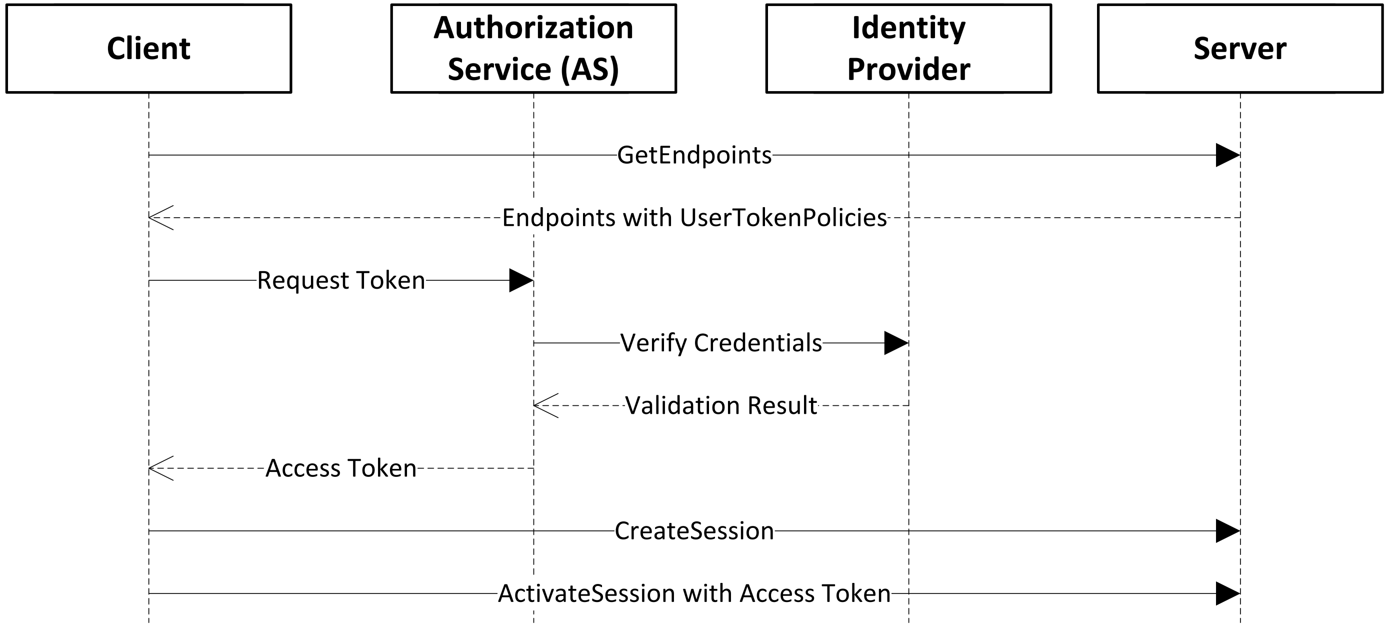
User Authentication

- User Credentials are tokens passed to the Server
- Different types of Token Types may be supported
- Security may be applied independently of the SecureChannel
- APIs exist to allow remote configuration of Servers.
- Access Rights granted based User Credentials AND Application Certificate

Token Type	Description	Security Requirement
Anonymous	No credentials supplied.	None
UserName	UserName with a password.	Encryption
X509	An X509 Certificate.	Signed
Issued Token (JWT)	A JWT issued by known authority.	Encryption

Authorization Services and Access Tokens

- Authorization Services allow Clients to request Access Tokens (JWTs);
- Identity Providers centralize management of User Credentials;
- Centralized management means individual servers do not need access to passwords.
- Access Tokens are specific to a Server
- Access Tokens expire



Roles And Permissions

- Roles are associated with a Session
- Permissions are associated with a combination of a Role and a Node.
- A Session will have access to a Permission if the one or more of its Roles has the Permission.
- Mapping Rules are used to determine which Roles are available for a Session

Role	Mapping Rules	Description
Anonymous	Identities = Anonymous Applications = Endpoints =	An identity mapping rule that specifies the Role applies to anonymous users.
AuthenticatedUser	Identities = AuthenticatedUser Applications = Endpoints =	An identity mapping rule that specifies the Role applies to authenticated users.

Operator1	Identities = User with name 'Joe'
	Applications = urn:OperatorStation1
	Endpoints =
		An identity mapping rule that specifies specific users that have access to the Role with a application rule that restricts access to a single Client application.

Operator2	Identities = Users with name 'Joe' or 'Ann'
	Applications = urn:OperatorStation2
	Endpoints =
		An identity mapping rule that specifies specific users that have access to the Role with a application rule that restricts access to a single Client application.

Supervisor	Identities = User with name 'Root' Applications = Endpoints =	An identity mapping rule that specifies specific users that have access to the Role

Administrator	Identities = User with name 'Root'
	Applications =
	Endpoints = opc.tcp://127.0.0.1:48000
		An identity mapping rule that specifies specific users that have access to the Role when they connect via a specific Endpoint.

Example of Assigning Roles to Sessions

Role	Mapping Rules
Anonymous	Identities = Anonymous Applications = Endpoints =
AuthenticatedUser	Identities = AuthenticatedUser Applications = Endpoints =

Operator1	Identities = User with name ‘Joe’
	Applications = urn:OperatorStation1
	Endpoints =

Operator2	Identities = Users with name ‘Joe’ or ‘Ann’
	Applications = urn:OperatorStation2
	Endpoints =

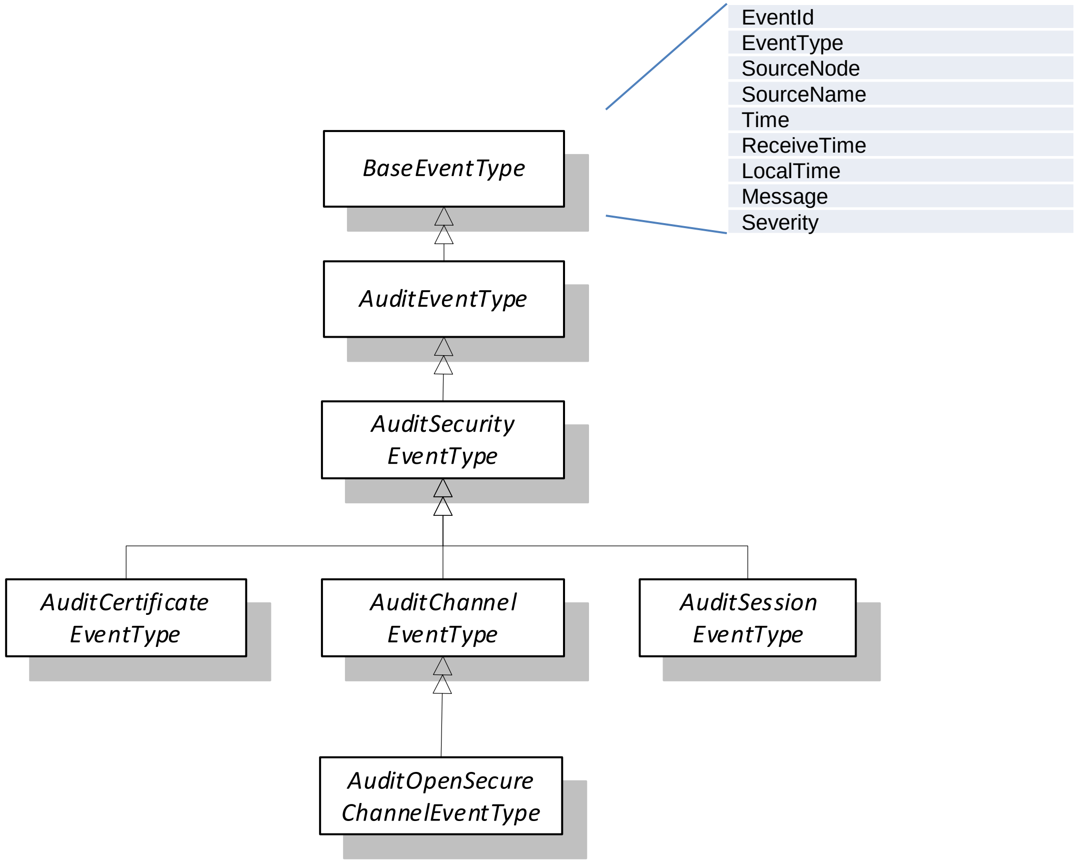
Supervisor	Identities = User with name ‘Root’ Applications = Endpoints =

Administrator	Identities = User with name ‘Root’
	Applications =
	Endpoints = opc.tcp://127.0.0.1:48000

User Provided by Client	Roles Assigned to Session
Anonymous	Anonymous
Sam	AuthenticatedUser
Joe using OperatorStation1 application.	AuthenticatedUser, Operator1
Joe using OperatorStation2 application.	AuthenticatedUser, Operator2
Joe using generic application.	AuthenticatedUser
Root using OperatorStation1 application.	AuthenticatedUser, Supervisor
Root using generic application and 127.0.0.1 endpoint.	AuthenticatedUser, Supervisor, Administrator
Root using generic application and another endpoint.	AuthenticatedUser, Supervisor

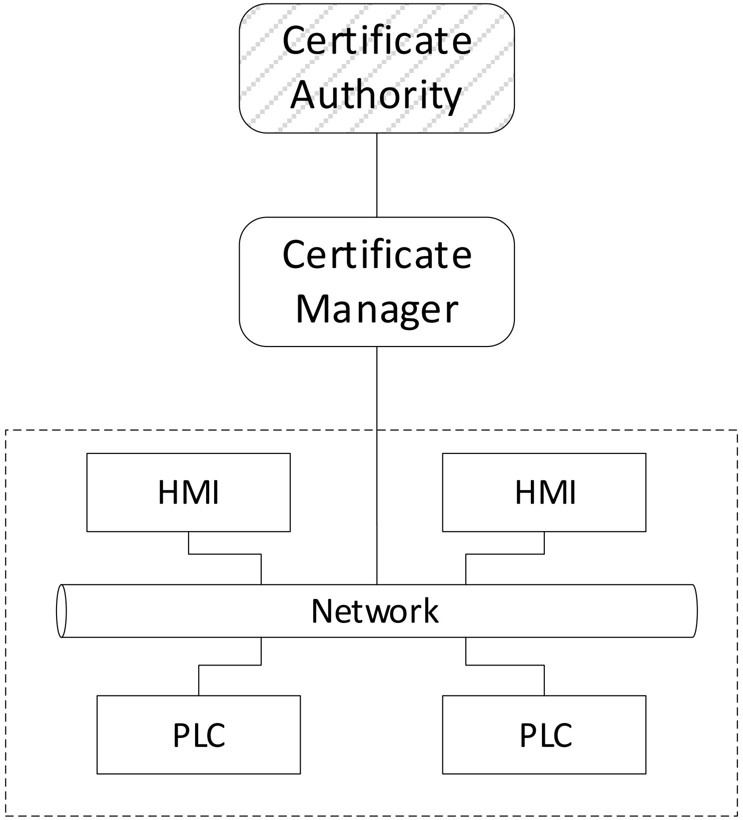
Auditing and Events

- AuditEvents are Events that are generated as a result of an action taken on the Server by a Client of the Server.
- AuditSecurityEvents are Events related to Security such as validating a Certificate or UserIdentityToken.
- AuditEvents may be reported vis Subscriptions, PubSub or via non-UA mechanisms such as SYSLOG.
- The structure and semantics of the events is the same no matter how they are reported.



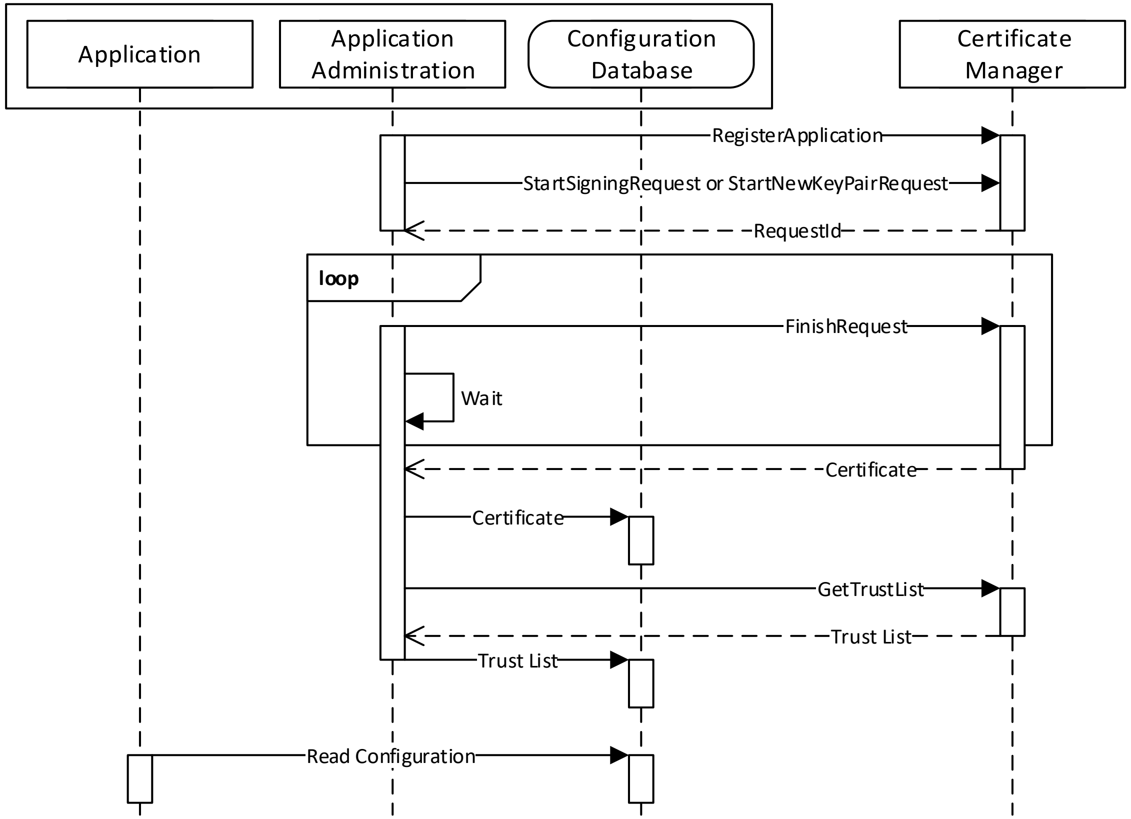
Certificate Manager

- A CertificateManager provides services to issuer and update Certificates and Trust Lists and to check CRLs.
- It is a front end to a Certificate Authority which maybe part of the corporate IT infrastructure.
- The CertificateManager makes it possible to centrally manage security for OPC UA applications.



Managing Certificates – Pull Mode

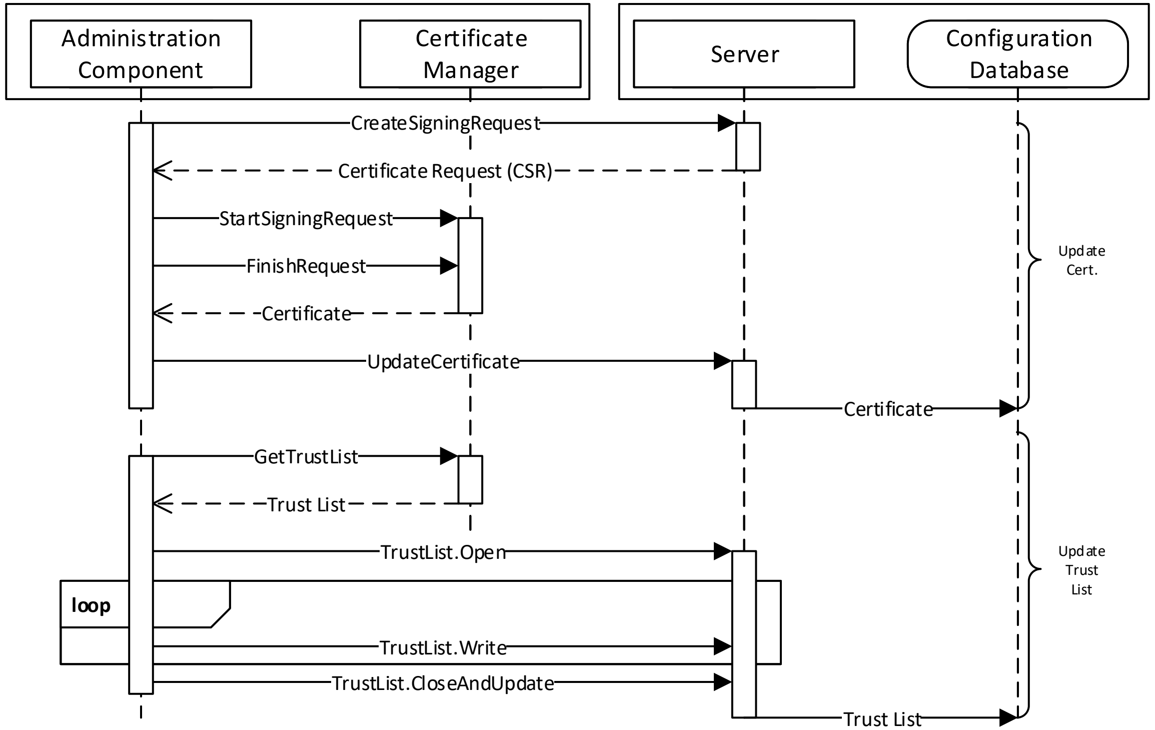
- Manual Configuration
  - Each application updates by Administrator
- Centralized Configuration
  - Use a Certificate Manager!
- Pull mode allows an Application to periodically update its won Trust Lists
- Push mode allows the Certificate Manager to push changes out to the applications.



Managing Certificates – Push Mode

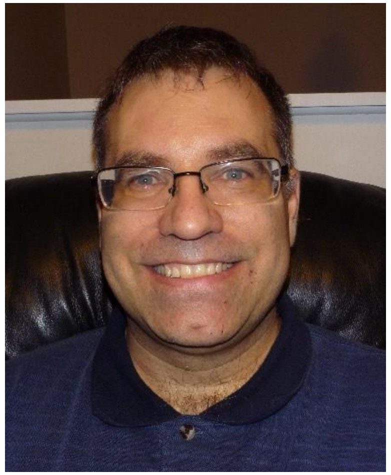
- The Push and Pull model also allow the initial certificate to be provided when approved by a human administrator.

- Applications can be removed by updating the CRL and pushing them out.



OPC UA Security Deep Dive

Questions?



Randy Armstrong
Chair of OPC UA Security Working Group
OPC Foundation
randy.armstrong@opcfoundation.org