Japan Computer Emergency Response Team Coordination Center

電子署名者 : Japan Computer Emergency Response Team Coordination Center
DN : c=JP, st=Tokyo, l=Chiyoda-ku, email=office@jpcert.or.jp, o=Japan Computer Emergency Response Team Coordination Center, cn=Japan Computer Emergency Response Team Coordination Center
日付 : 2011.02.18 15:48:19 +09'00'

OPC UAとUA Securityのご紹介

制御システムセキュリティカンファレンス2011
2011.2.10
コクヨホール

日本OPC協議会
発表者:　藤井　稔久

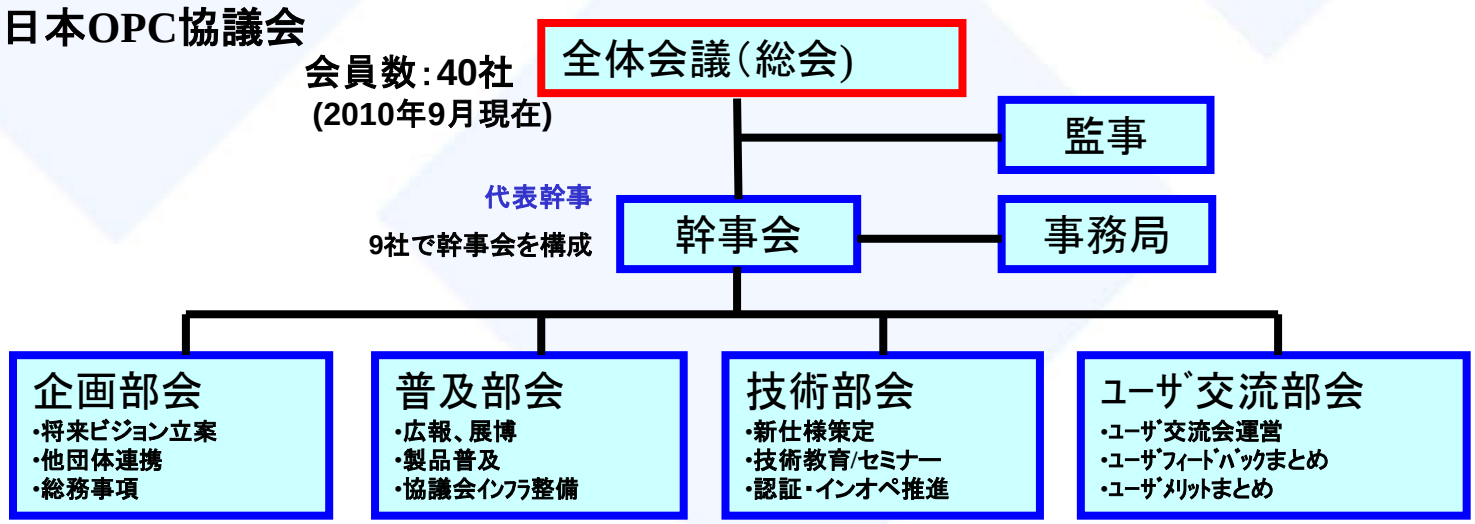
目次

- OPC協議会について
- UA仕様について
- UAセキュリティについて
- 参考情報 ： UAシステムが想定する脅威と対策

OPC Foundation / OPC協議会

日本OPC協議会

- 国際業界標準化団体
  - 加入会社 392社(エンドユーザ 60社)　　※2010/10/8現在
  - 3,500を超えるOPC製品取り扱い会社 = 22,000を 超えるOPC製品
  - 数百万規模のインストールベース
- OPCのビジョンは、マルチベンダー、マルチプラットフォームにおけるセキュアで高信頼な相互運用環境を提供すること
  - データソースから基幹業務系における情報の垂直統合
  - 異なるベンダーの異なるネットワーク間における情報の水平統合
  - 単なるデータとしてではなく、情報として……
- 高信頼でセキュアな相互運用性は必須要件(オプションではない)
- 統合オープンプラットフォームアーキテクチャにより、各種のオープンな標準(国際標準、業界標準など)の連携を実現



UA仕様について

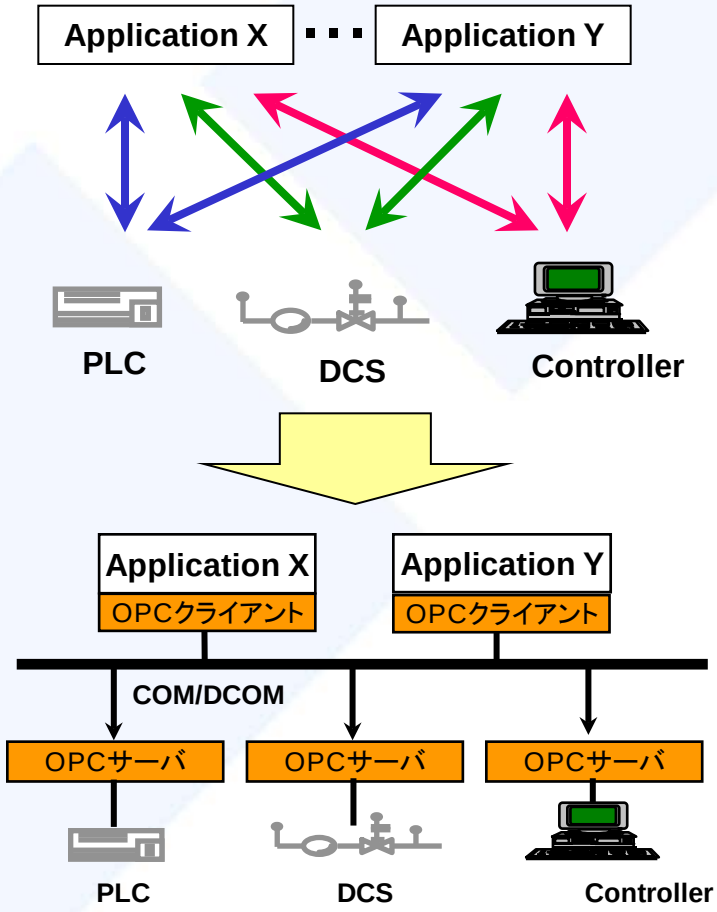
Classic OPC ： プロセスデータ交換 I/F

日本OPC協議会

OPC

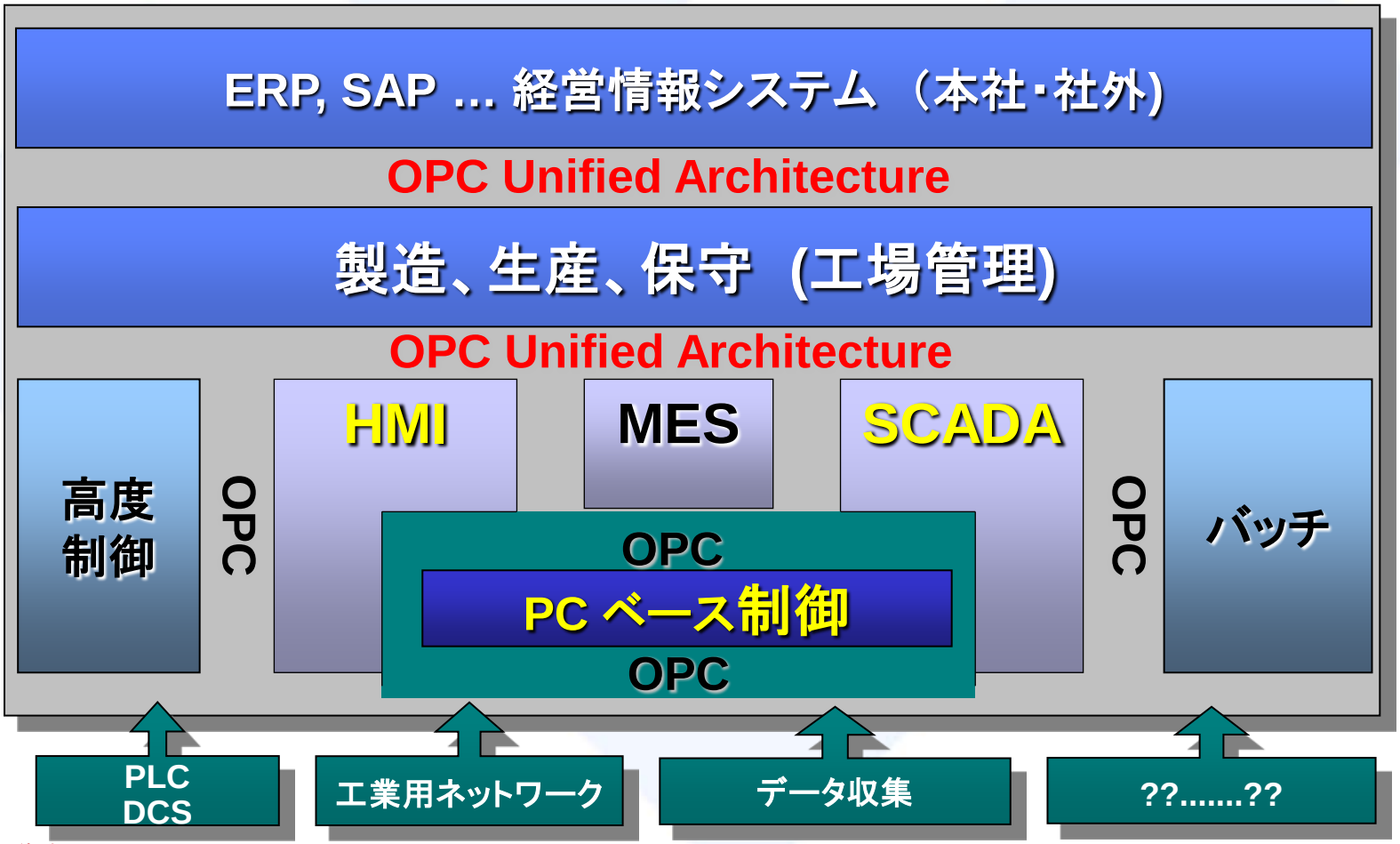
⇒ ベンダや機器に依存することなくデータ交換を行なうClient / Server システムのための標準インタフェースです。

- **課題**
  - ✓ 多数のベンダー製品
  - ✓ カスタムメイドのソリューション
  - ✓ プロプラエタリィな技術
  - ✓ 1対1結合による統合
  - ✓ 限られたリアルタイム情報
  - ✓ 保守の悪夢
  - ✓ 散逸した責任
- **解決策**
  - ✓ OPC！

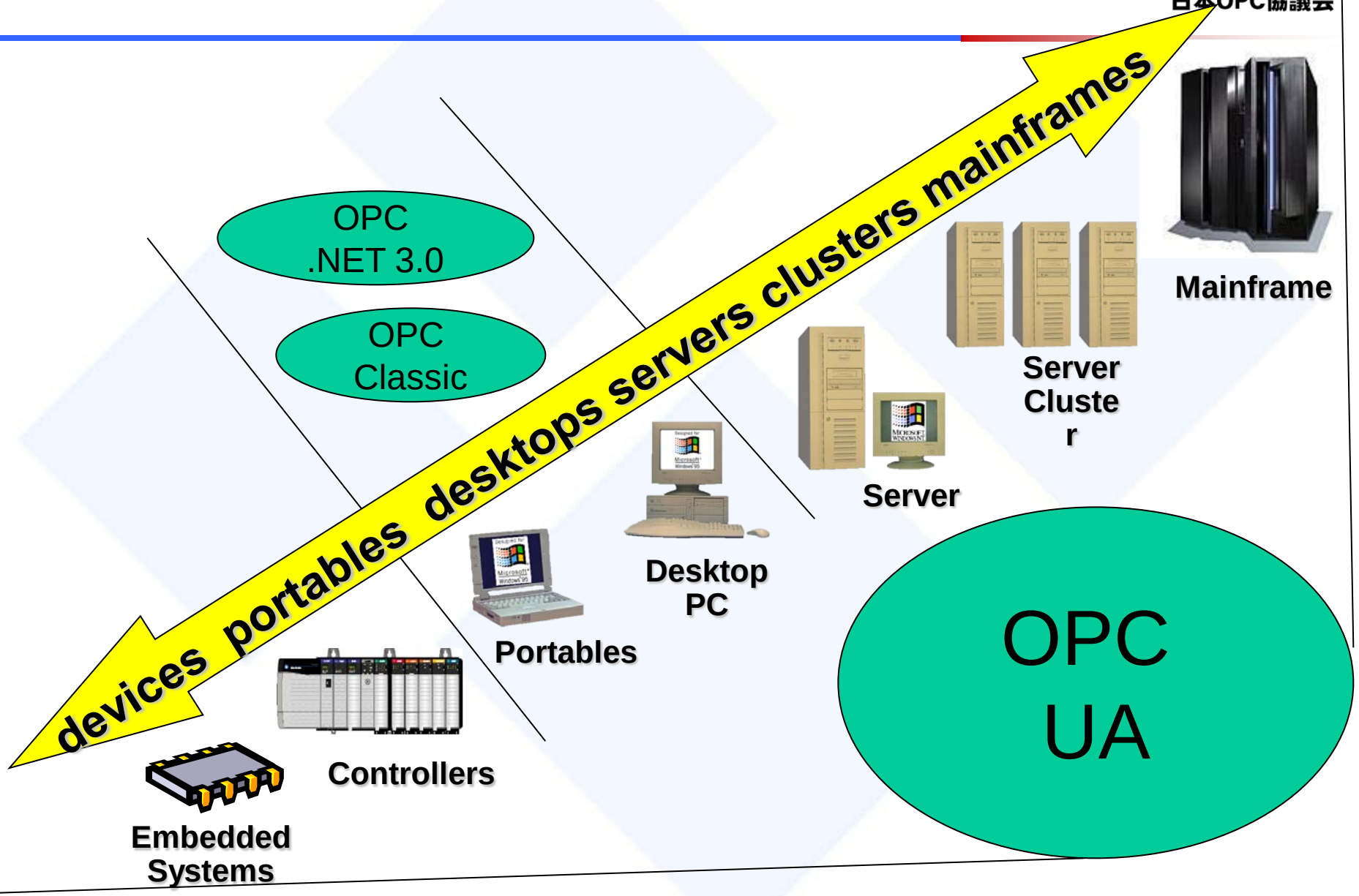


OPC UA： 製造情報データ連携 I/F

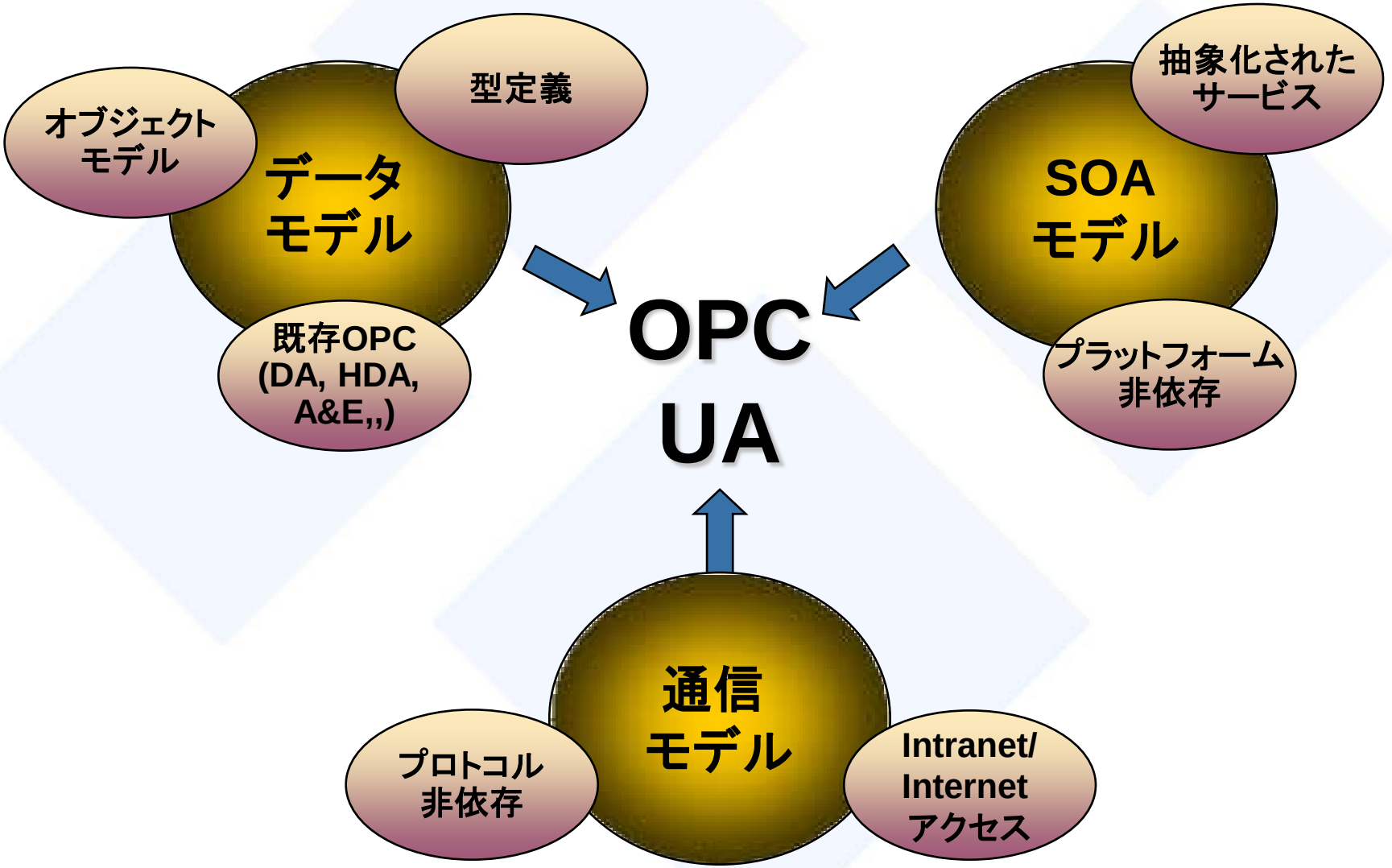
OPC UA： 業界標準の相互運用性を提供
⇒ interOperability, Productivity & Collaboration



OPC スケーラビリティ



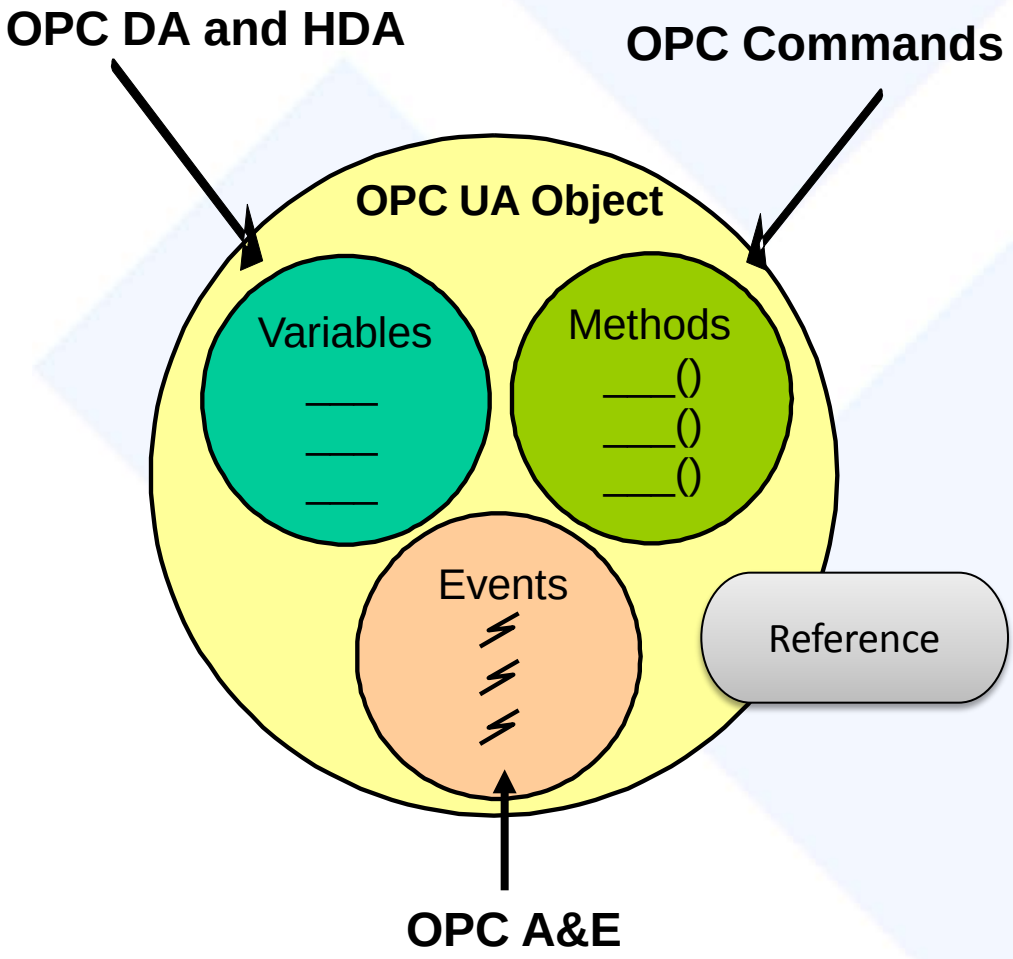
特徴：UA (Unified Architecture) とは？



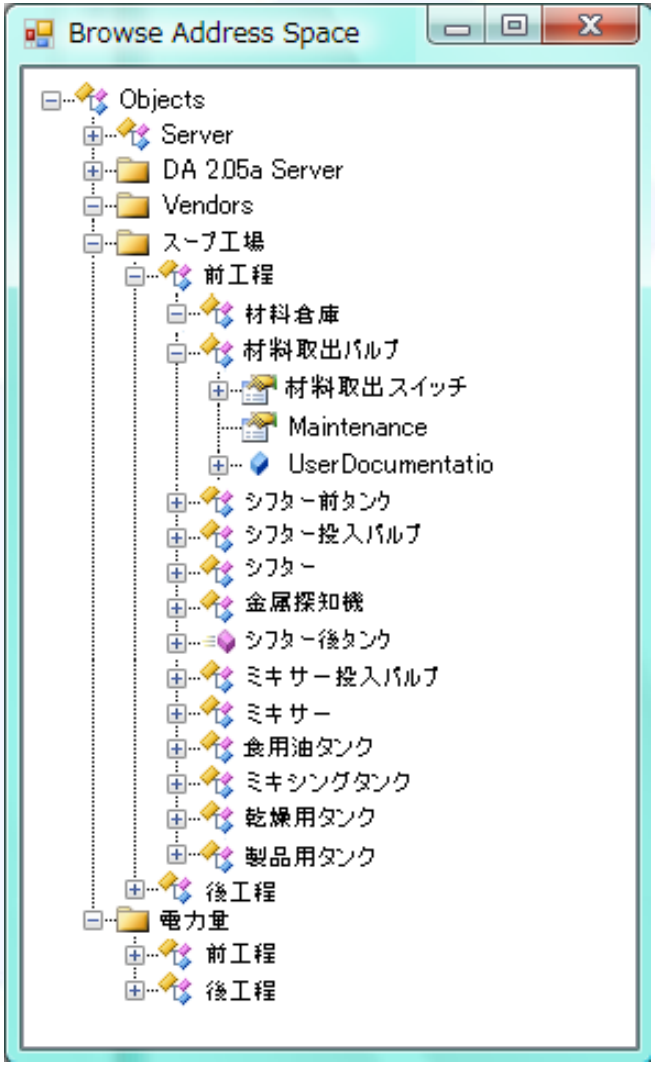
オブジェクトモデル&アドレススペース

日本OPC協議会

UA オブジェクトモデル



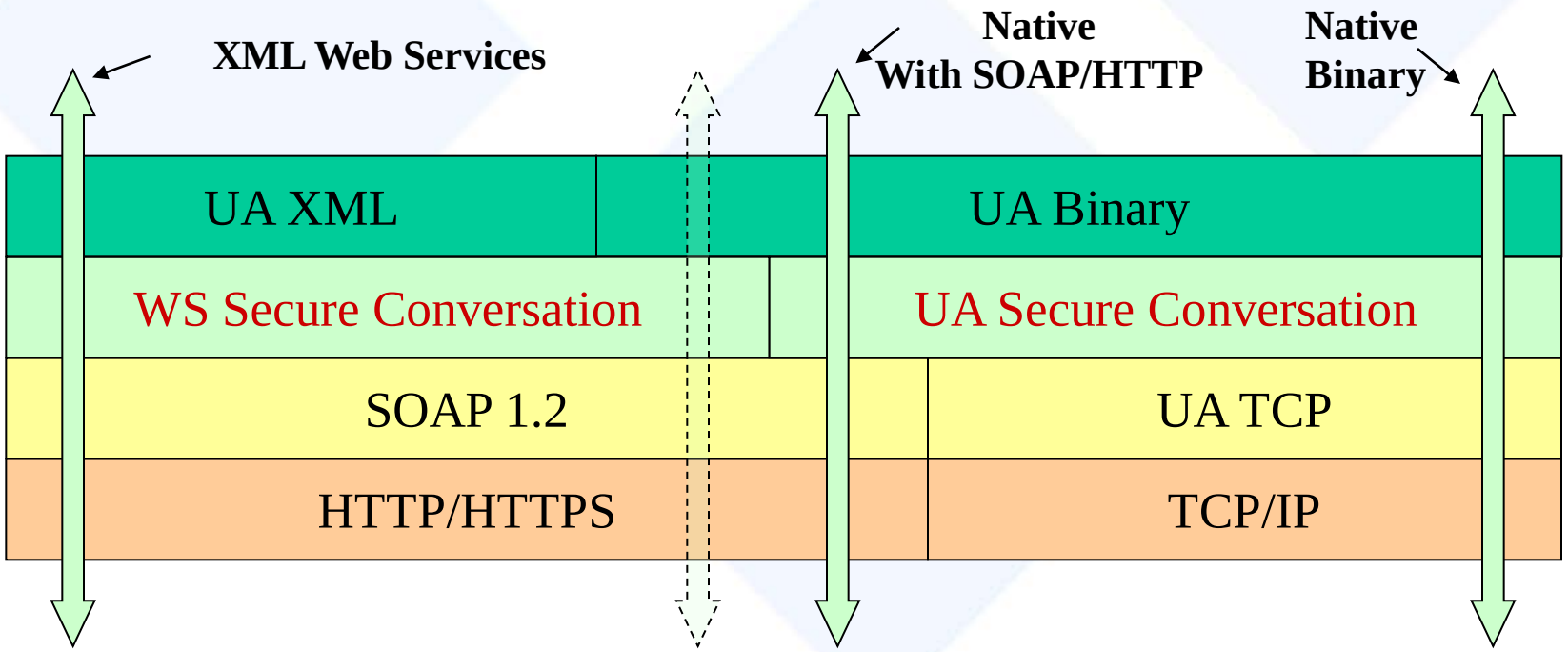
UA アドレススペース



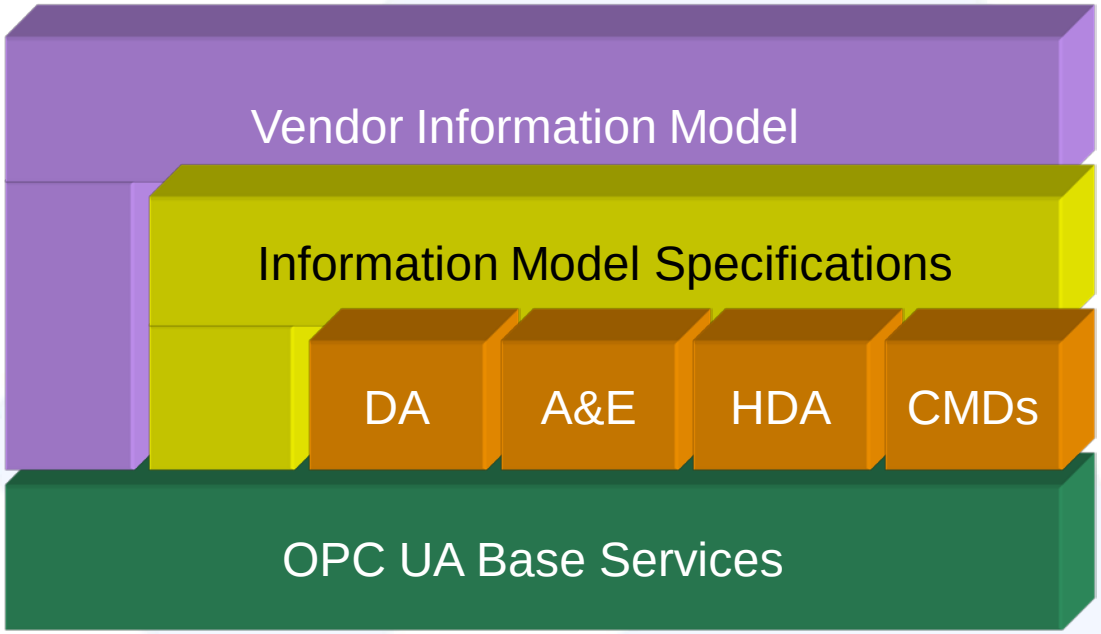
スタック プロファイル

SecurityをBuilt-inした3種類のスタックプロファイル:

- XML Web Service　　: ITシステムとの親和性⇒ ERP / ビジネスApp
- Native Binary　　: 高速通信⇒組み込み / PCベース制御 / SCADA
- Native with SOAP/HTTP :　MES / ERP



インタフェース レイヤ



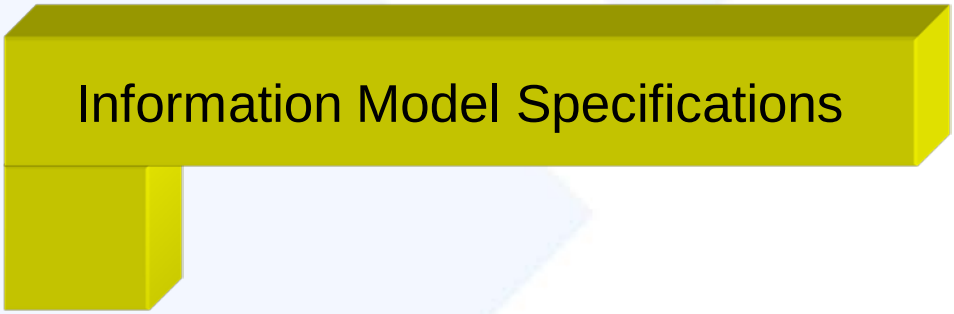
FDI, ADI, PLCopen

OPC Information Model

All Necessary Services

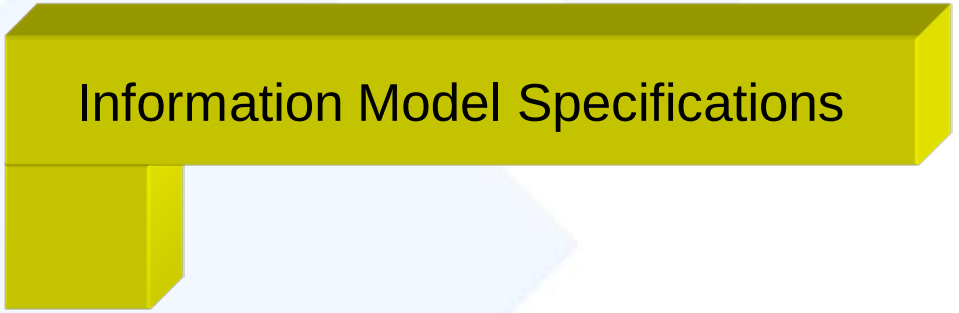
活動中の協業プロジェクト

- *EDDL + FDT = FDI*
- *ADI*
- *PLCopen*



検討中の協業プロジェクト

- *Building Automation*
- *OpenO&M*
- *Smart Grid*
- *MTConnect*
- *ISA-95*
- *ISA-88*



UA Securityについて

UA Security

- UAセキュリティ:仕様書(12部)のコア仕様として定義
  - Part.1 :Concepts & Overview //概要
  - ----
  - **Part.2 :Security Model //コア仕様**
  - **Part.3 :Address Space Model // ..**
  - **Part.4 :Services // ..**
  - **Part.5 :Information Model // ..**
  - **Part.6 :Service Mappings // ..**
  - **Part.7 :Profiles // ..**
  - ----
  - Part.8 :Data Access //OPC独自情報モデル
  - Part.9 :Alarms and Conditions // ..
  - Part.10 :Programs // ..
  - Part.11 :Historical Access // ..
  - ----
  - Part.12 :Discovery

- UA Security：目的･脅威･対策

日本OPC協議会

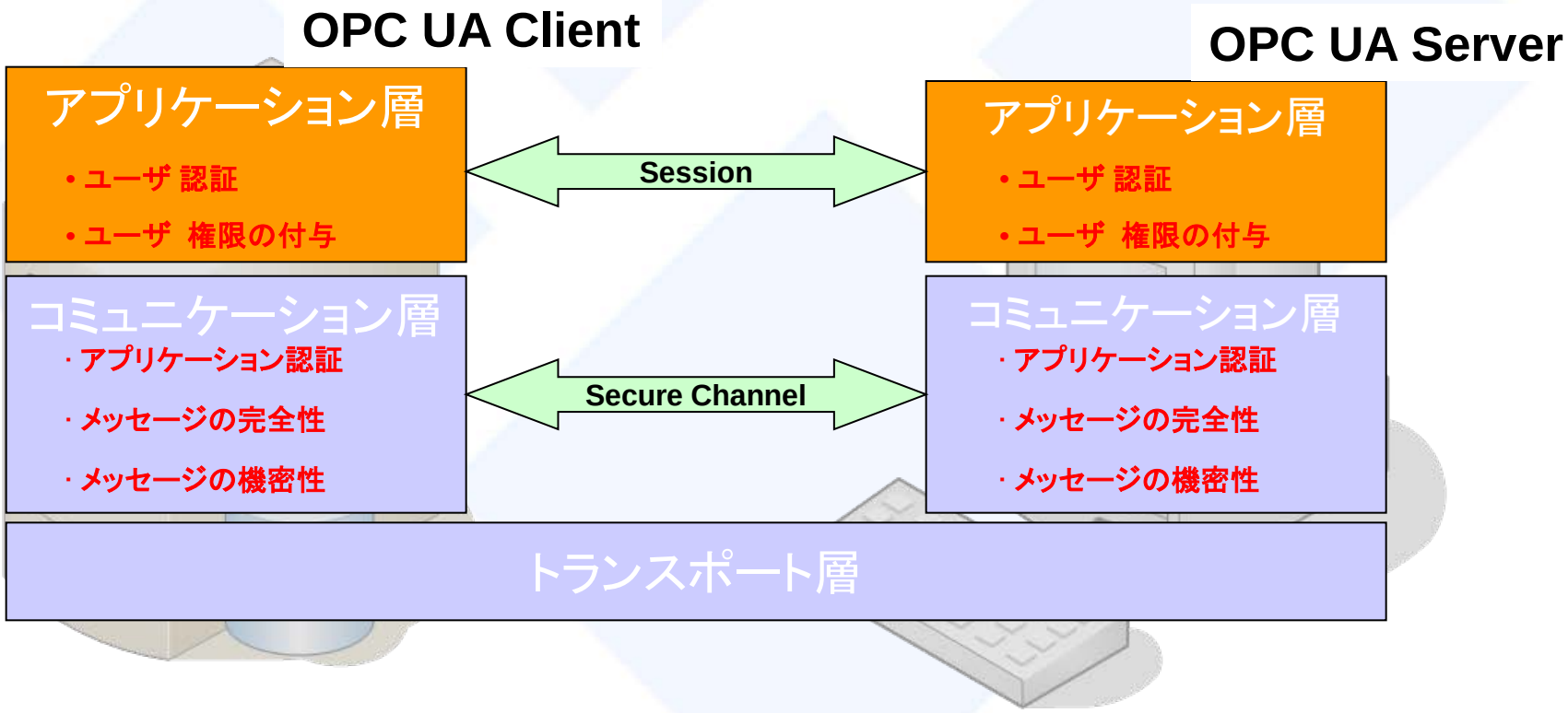
Part.2：UAに求められるクライアント/サーバシステムのSecurityアセスメント結果

Threat(s)	Objective(s)
･Message Flooding ･Eavesdropping (盗聴) ･Message Spoofing(ﾒｯｾｰｼﾞ捏造) ･Message Alteration(ﾒｯｾｰｼﾞ改竄) ･Message Replay ･Malformed Message ･Server Profiling ･Session Hijacking ･。。。	● Auhtentication ● Authorization ● Confidentiality ● Integrity ● Auditability ● Availability

詳細は後述の参考情報を参照

- UA Security : アプリケーション

- Secure アプリケーション
  安全なアーキテクチャを採用し、⇒安全メカニズムをビルトイン



- UA Security :プロファイル（Part.7)

日本OPC協議会

- Security モード
  - None – セキュリティなし
  - Sign – メッセージに署名は付けるが、暗号化はしない。
  - SignAndEncrypt –メッセージに署名は付けかつ暗号化する。
- Security ポリシー
  - Basic128Rsa15 – 最低限のセキュリティ(⇔高速応答性が必要な場合)
  - Basic256 – 推奨セキュリティ
  - None – 推奨できない

⇒ Security要件に応じて組合せをプロファイルに定義し、使い分けます。

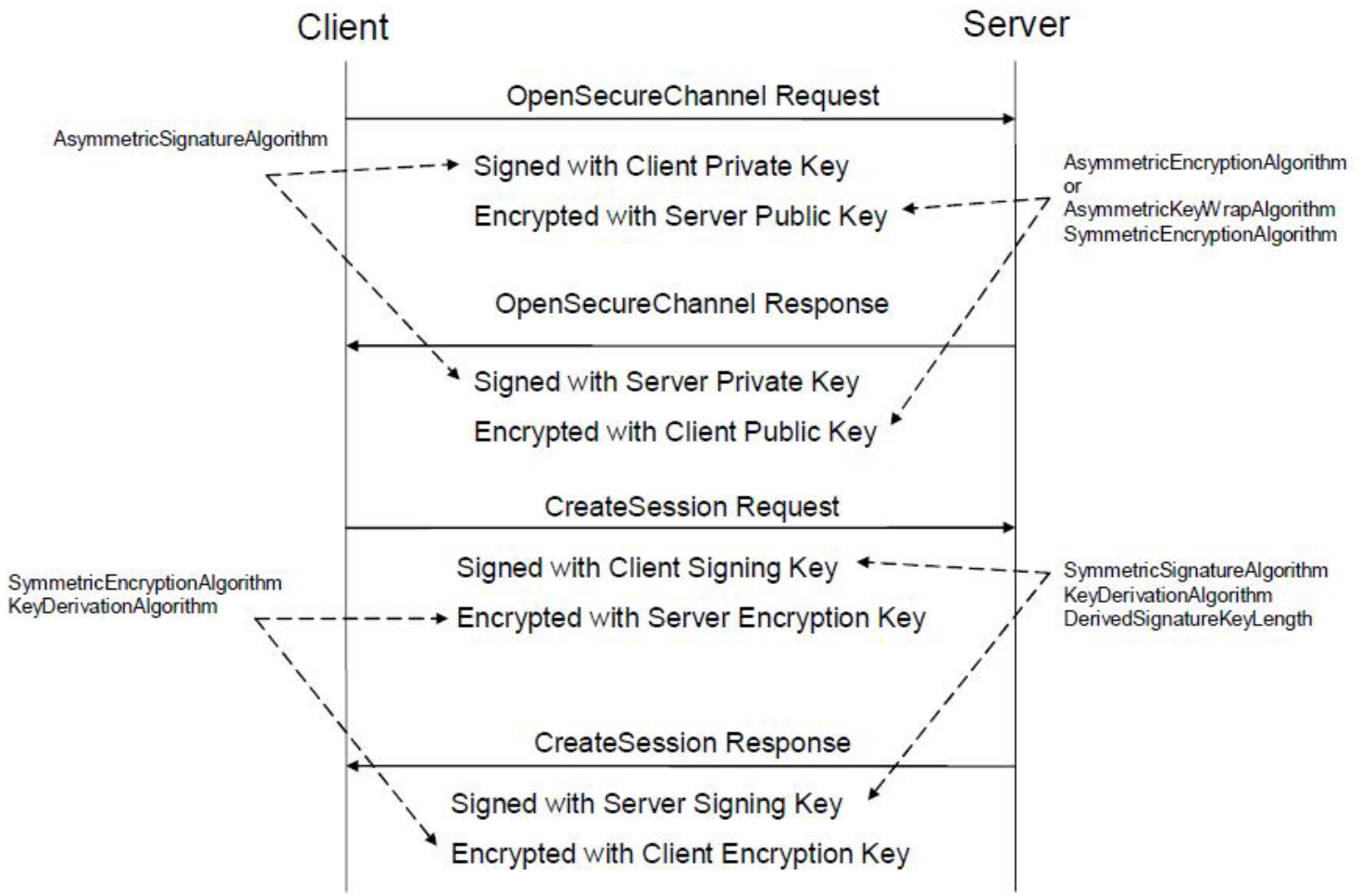
- UA Security: サービス（Part.4)

日本OPC協議会

- セッション
  - UAアプリケーション層で実装
  - Client/Server間のハイレベルな論理的接続機能を提供
  - ユーザの認証・認可(アクセス制御)によりClientのServerアクセスを制限
  - セッションの確立にはSecureチャネルの開設が必要
  - Secureチャネルが切断しても、セッションは維持したままでの再接続が可能

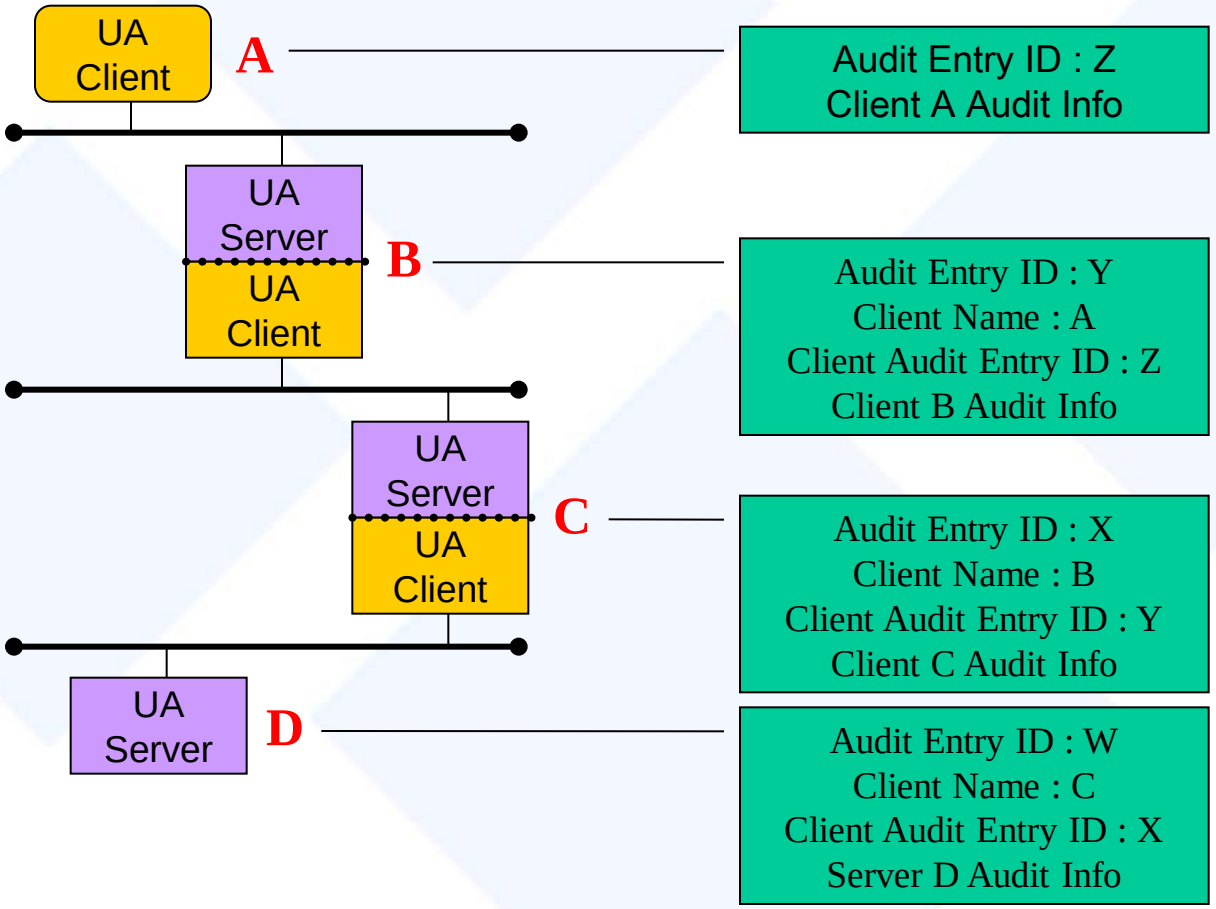
- Secureチャネル
  - UAコミュニケーション層(通信スタック)で実装
  - Client/Server間のローレベルな論理的接続機能を提供
  - アプリケーション認証のための証明書を提供
  - 送信メッセージを暗号化するなど、Sercureなメッセージ送信を可能に！
  - 受信メッセージの署名による改ざん検査(Verify)など、Sercureなメッセージ受信を可能に！
  - Secureチャネルを維持したままでのトランスポート層の再接続が可能

- UA Security: Secureチャネル



OpenSecureChannelによるセッション確立時は非対称方式で鍵共有を確立。
その後のメッセージ交換はOpenSecureChannelで交換した乱数（nonce）から派生した鍵を使用

- UA Security: 監査



- UA Security : まとめ

①UAセキュリティは特殊なものではありません。

⇒ITの世界で標準的に使用されているセキュリティ対策を使用します。

②UAセキュリティのための特別な実装は不要です。

⇒スタックに実装されている機能を利用することにより
その恩恵を簡単に享受することができます。

参考情報:

UA システムが想定する脅威と対策

- UA Security：目的

Authentication （認証）

なりすまし などへの対策。　対象が本当に正しいものだと証明する。

- UAの対策
  - Application Authentication (クライアントとサーバーのアプリケーション)の識別に http://tools.ietf.org/html/rfc3174 X.509 Certificatesを用いている。
  - User Authenticationに、使用プロトコルに応じて username/password, X.509.v3 certificate, WS-SecurityToken を用いている。
    - User Identity TokenがDigital Certificateである場合、このトークンはchallenge-response process（サーバーは乱数と署名アルゴリズムをCreateSessionの返信で渡し、クライアントはサーバ発行乱数に署名を施したものをActivateSession以降のサービス呼び出しの引数として渡す処理手段）で評価される。

- UA Security：目的

Authorization （権限）

正しい権限を持ったものにしか適切な処理は行わせない。

- UAの対策
  - アプリケーション作成側の処理に委ねる。アプリケーション作成側はAuthentication情報を参照することが出来る。

- UA Security：目的

Confidentiality（信頼性）

盗聴　などへの対策。

- UAの対策
  - 暗号化
    - 非対称鍵暗号（asymmetric encription　公開鍵で暗号化して秘密鍵で複合化する。）を鍵共有(key agreement)に用いる。
    - 対称鍵暗号（symmetric encription　暗号化と複合化に同じ鍵を使う。）を他のメッセージ交換に用いる。

- 制限
  - ネットワークやシステム構造のConfidentialityはCSMS(Cyber Security Management System)に依存する。
  - 鍵の管理には公開鍵基盤（Public Key Infrastructure 利用者の身元について「信頼できる第三者」が審査を行い、保証を実現する仕組み)に依存する。

- UA Security：目的

日本OPC協議会

Integrity（完全性）

改ざん　などへの対策。

- UAの対策
  - 署名
    - 非対称署名（asymmetric signature）をSecure Channel確立時の鍵共有（key agreement）に用いる。
    - 対称署名（symmetric signature）を他のメッセージ交換に用いる。
- 制限
  - ネットワークやシステム構造のIntegrityはCSMS(Cyber Security Management System)に依存する。
  - 鍵の管理には公開鍵基盤（Public Key Infrastructure 利用者の身元について「信頼できる第三者」が審査を行い、保証を実現する仕組み)に依存する。

- UA Security：目的

Auditability（監視）

セキュリティ状態が有効であるかを確認する為にシステムの運用状況を確認する。

- UAの対策
  - 通信スタックでは標準的な処理に対するAuditイベントを発行している。
  - Auditイベントの追加、Auditイベントを受信してどのよう処理するかはアプリケーション作成側の処理に委ねる。

- UA Security：目的

Availability （可用性）

メッセージ洪水　などへの対策。

- UAの対策
  - メッセージが認証される前の処理を最小化して、Message Floodingの影響を少なくする。
    - GetEndpointsとOpenSecureChannelサービスが唯一のユーザ認識される前にサーバが処理するもので、それ以外は認証されないメッセージは破棄。
    - GetEndpointsは静的情報を返すだけなのでサーバー側に負荷はない。
    - OpenSecureChannelについては1)不正な情報を受けたら意図的に処理を遅らせてアラート通知する、2)サーバ最大許容チャネル数を超過した接続には署名や暗号化処理を行わずにエラーを返す　という2つの仕組みでMessage Floodingに対応。
  - Authenticationにより、適正ユーザがMessage Floodingに利用されるリスクを低める。
  - Auditingにより、異常状況を認識させる。
- 制限
  - システムとネットワークプロトコルのレベルでのMessage FloodingについてはCSMS(Cyber Security Management System)に依存する。

- UA Security : 脅威

Message Flooding

クライアントがセッションを持っていないケース：UAのセッション確立能力を阻害したり、サーバー内にある既存セッションを終了させる。

クライアントがセッションを持っているケース：仕様外のメッセージを沢山送ってサーバリソースを枯渇させる。

通信を妨害してサービスを正常に実行させない。

- 阻害される目的 ： Availability
- UAの対策
  - Availability に記載した対策と同じ。

- UA Security : 脅威

Eavesdropping

盗聴

- 阻害される目的 ： Confidentiality、 間接的には他全てに影響し得る
- UAの対策
  - 暗号化

Message Spoofing

メッセージの捏造をしてクライアントやサーバに送る

- 阻害される目的 ： Integrity、Authorization
- UAの対策
  - メッセージ署名
  - SessionID, SecureChannelID, Timestamp, シーケンス番号, Request ID をメッセージに持ち正当性を評価している

- UA Security : 脅威

Message Alteration

メッセージ改ざん

- 阻害される目的 ： Integrity、Authorization
- UAの対策
  - メッセージが変更されている場合は署名データの検証により変更を認識し、そのメッセージを破棄

Message Reply

メッセージを別のタイミングで再送して誤った動き（不必要な時にバルブを開くなど）を引き起こす。

- 阻害される目的 ： Authorization
- UAの対策
  - Message Spoofingと同じ。
    - 同じメッセージを送ってもシーケンス番号が正当性を失っている。
    - データを捏造して同じメッセージを送っても署名データで検証される。

- UA Security : 脅威

Malformed Message

不正な構造のメッセージを送り想定外の処理を行わせる。サーバクラッシュなどに繋げる。

- 阻害される目的 ： Integrity、Availability
- UAの対策
  - UAのサービス呼び出しでは、メッセージが正しい様式と適正範囲内のパラメータ値になっているかをチェックしている

Server Profiling

相手に正しい/不正なメッセージを送り、その反応などからサーバーの情報を推定する。乗っ取りなどに繋げる。

- 阻害される目的 ： 全て
- UAの対策
  - 識別されていないクライアントに提供する情報は最小限にしている。
    - GetEndpointsの戻り値のみがその対象。他はクライアント識別済み。

- UA Security : 脅威

Server Hijacking

接続中セッションの情報を取得・類推してセッションを乗っ取り、任意のメッセージを送る。

- 阻害される目的 ： 全て
- UAの対策
  - 各セッションにセキュリティ情報を割当てている。
    - セッションのハイジャックには最初にそのセキュリティ情報に準拠する必要がある。

Rogue Server

偽UAサーバーをセットアップして情報入手する。

- 阻害される目的 ： Integrity以外の全て
- UAの対策
  - サーバのApplication Instance Certificationを用いる。
    - 偽サーバーが同じCertificationを持っていたとしても、暗号化技術のPrivate Keyが複製できない為に、正しいPublic Keyで守られたメッセージを利用できない。

- UA Security : 脅威

Compromising User Credentials

ユーザ情報（ユーザ名、パスワード、認証、キーなど）を取得して悪意にサーバにアクセスする。

- 阻害される目的 ： Authorization, Confidentiality
- UAの対策
  - ユーザ情報の暗号化
- 制限
  - パスワード類推やソーシャルエンジニアリングに対してはCSMS(Cyber Security Management System)に依存する。

Questions?

- 日本OPC協議会