SECURE INTERNET TECHNOLOGIES

Xtradyne

Security for Parlay-X

challenges and solutions

Tim Eckardt

Xtradyne Technologies AG

eckardt@xtradyne.com

www.xtradyne.com

Parlay Group Open Meeting – Rome,
November 4-6, 2003

Outline

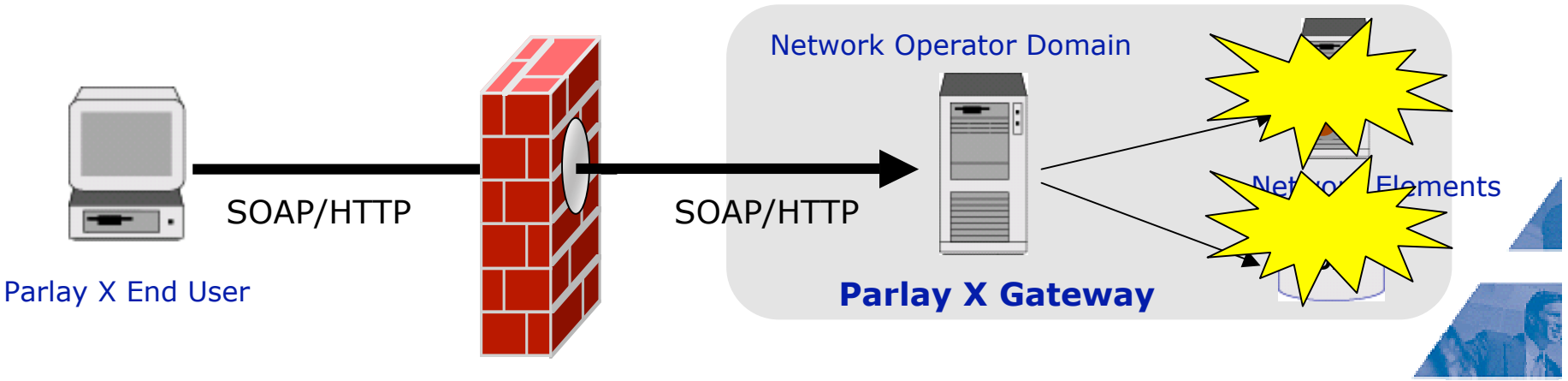
- **Web Services: A Paradigm Change**
  - brief overview from a security perspective
  - security risks and problems
- **Web Services Security Standards**
  - emerging security standards for XML, SOAP, & Web services
- **Available Security Solutions & Best Practices**
  - WS-security toolkits
  - SOAP firewalls, application firewalls for XML/SOAP/Web services

Web Services – a paradigm change

- **XML-based applications**
  - modular, extensible, service-oriented interfaces
  - Internet protocols (HTTP)
  - ASCII-based transfer syntax
- **Loose coupling**
  - SOAP appropriate for inter-application communications:
    - **asynchronous** vs synchronous
    - few, **coarse-grained** (**service-oriented**) interfaces vs many fine-grained (object-oriented) interfaces
    - **extensible** specifications vs tightly-coupled implementation dependence
- **Document-based exchange patterns**
  - self-contained messages w/o connection-based context!
  - per-message security context vs per-connection security context
  - ⇒ SSL is of limited use only!
    - XML Encryption, XML Signature more applicable

Web Services – an inherent risk?

- **Integration**
  - new access paths and data flows exposure
  - exposure of critical network resources risk
- **Web Services are firewall-friendly ;-)**
  - HTTP used as firewall-friendly transport layer for SOAP msgs
- **Web Services are *too* firewall-friendly*!***
  - HTTP increasingly used as universal, firewall-outwitting tunnel!
  - HTTP not properly filtered by most standard firewalls!
  - SOAP not filtered at all by standard firewalls!



Risks, Threats, and Challenges

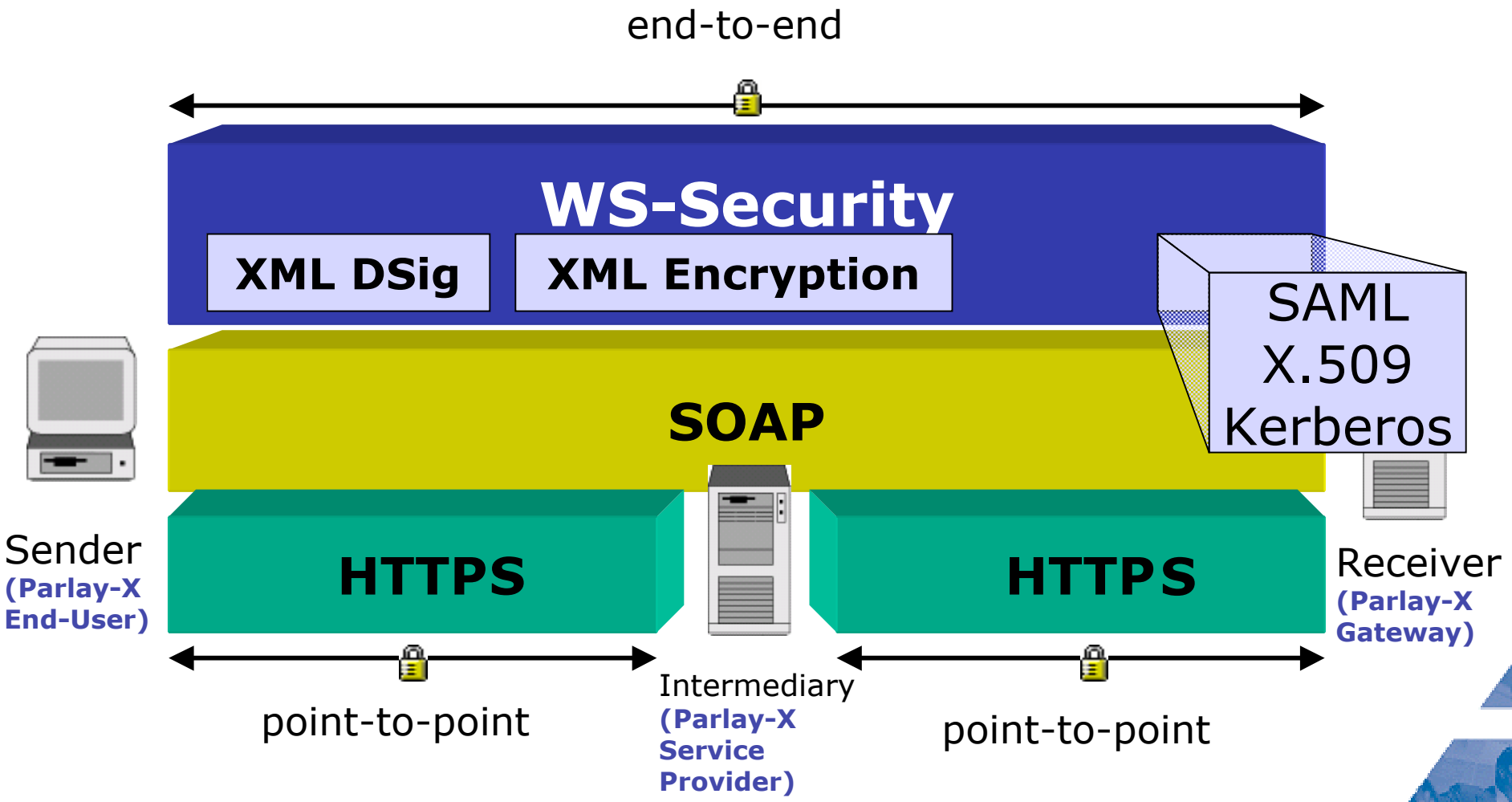
Potential attacks:

- **… on Parlay-X messages**
  - *Eavesdropping* - privacy breaches, fraud, espionage
  - *Modification* - sabotage, fraud
  - *Fabrication, Replay* - sabotage, fraud
  - *Drop, Redirect* - sabotage, fraud
- **… on Parlay-X services & gateways**
  - *Unauthorized Access* - theft, fraud, sabotage, espionage
  - *Tampering, Denial of Service* - sabotage, staging of further attacks
- **techniques used for staging attacks**
  - at transport level: sniffing (ethereal), netcat, TCP-hijacking
  - at application level: application-specific attacks (e.g., SQL injection)
  - . . .

Risks, Threats, and Challenges - 2 -

- **Loose coupling**
  - Web Services are message-based, (self-contained msgs)
  - ⇒ transport-layer security sessions (SSL sessions) do not fit anymore!
- **HTTP transport**
  - SOAP messages pass firewalls without inspection
  - ⇒ existing perimeter protection does not help
- **Service composition**
  - a single SOAP message can traverse many intermediaries
  - ⇒ who can you trust with what?
- **Document-centric workflows**
  - different parts of a SOAP message are:
    - created/inserted by various parties
    - read/processed by different SOAP processors
    - need different levels of security

Web Services Security Standards



Web Services Security Standards - 2 -

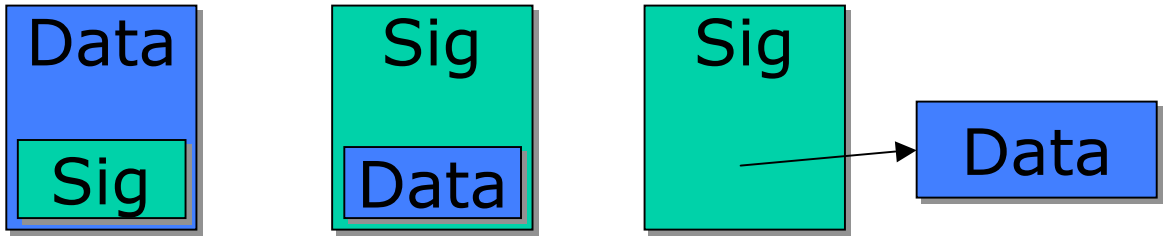
WS-Security:

- **OASIS-Standard**
  - Working Draft since 11/2002
- **Message-level Security Model for SOAP**
  - can embed a wide variety of existing technologies
  - end-to-end security with multiple trust domains possible
- **Extensible security message header** `<wsse:security>`
  - for security information *in* and *about* messages
- **Security Token format**
  - express claims (assertions) made by various entities
  - text/binary, signed/unsigned, (e.g. username, X.509 certificates, Kerberos tickets)
- **Integrity, Authentication, Confidentiality**
  - processing rules for XML Digital Signature and XML Encryption
- **Common framework for future specifications**
  - WS-Policy, WS-Trust, WS-Federation, …

Web Services Security Standards - 3 -

XML Digital Signature + XML Encryption:

- **W3C-Standards**
  - "Recommendations" since 2002
- **XML Syntax for signatures and encrypted data**
  - *partial & persistent* signing/encrypting of XML content
  - not just for signing/encrypting XML content!
  - no new algorithms or protocols
  - signatures: *enveloped, enveloping, detached*



- **Usage in WS-Security:**
  - integrity/confidentiality protection for *individual* parts of a message (header, body, attachments)
  - authentication of security tokens
  - binding security tokens to messages

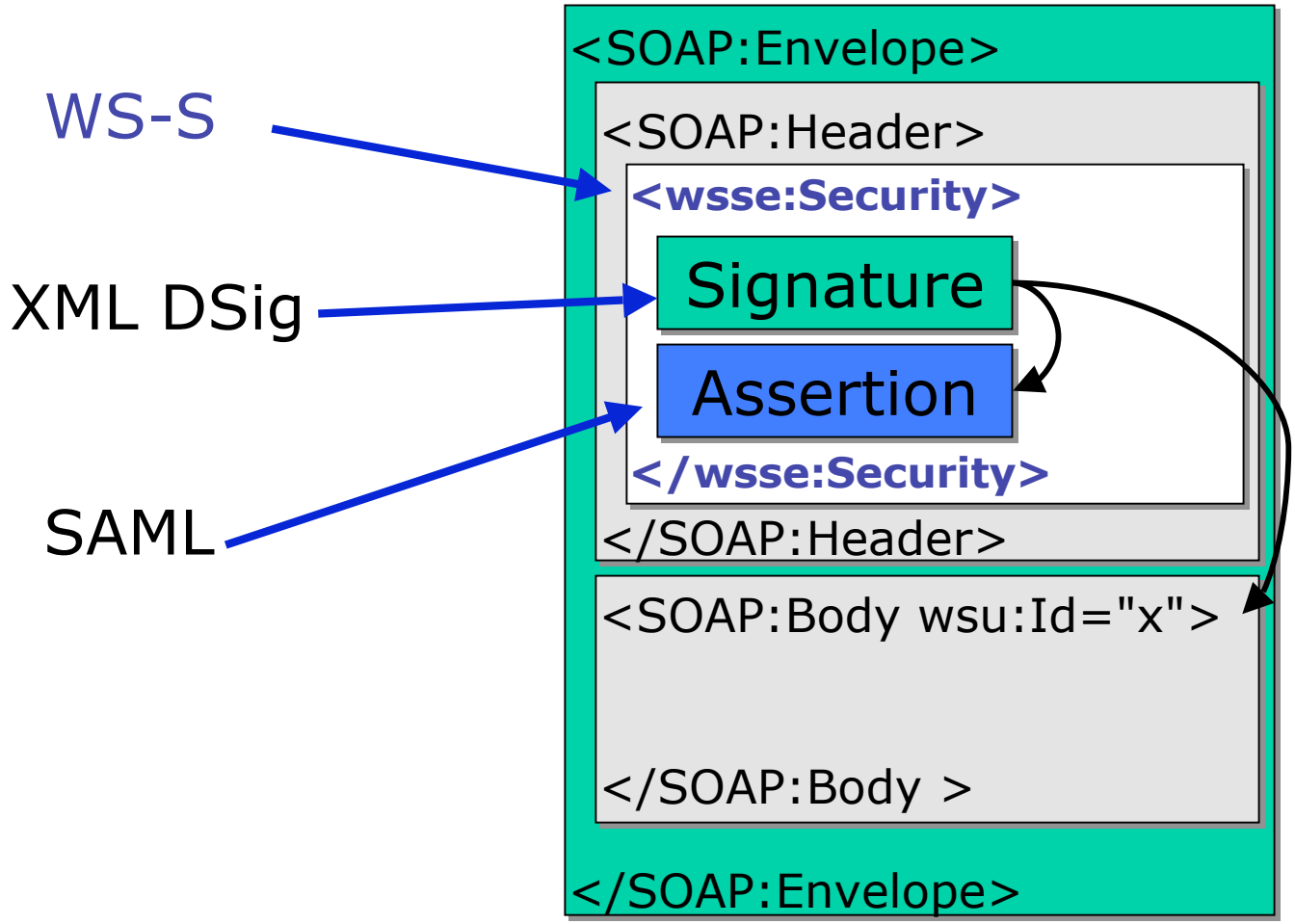
Web Services Security Standards - 4 -

Security Assertion Markup Language (SAML):

- **OASIS-Standard**
  - SAML 1.0 since 5/2002
  - SAML 1.1 since 9/2003
- **XML-based framework**
  - for the exchange of security information (*Assertions*)
  - Assertions = statements by an issuer about a subject
    - Authentication Assertion
    - Authorization Assertion
    - Attribute Assertion
- **SAML Protocol**
  - request/response protocol messages between Policy Enforcement Points and Policy Decision Points
- **Usage of SAML Assertions in WS-S**
  - SAML Assertions as format for Security Tokens
  - Binding to WS-Security in progress
    - "*SAML Token Binding*"

Web Services Security Standards - 5 -

Security standards in concert:



Available Security Solutions

- **WS-Security toolkits**
  - Security implemented *as part of the application at the end-system*

- **WS-Security Gateways/ SOAP Security Proxies**
  - aka ***'SOAP Firewalls'***
  - aka ***'WS-Domain Boundary Controller (WS-DBC)'***

Available Security Solutions - 2 -

WS-Security toolkits:

- **Security implemented *as part of the application***
  - implementation (coding) needed in *every single* end-system
  - toolkits available for some security functionality
  - resource-intensive (CPU cycles) processing needed at Parlay-X Gateway
    - severe impact on service-handling capacity of Parlay-X Gateway
    - denial-of-service by heavy loads of requests for execution of security mechanisms (authentication, encryption, dig. signature)
  - WS-Security standardization cycles and application release cycles must be coordinated

- **Drawbacks**
  - security and application code mixed
  - security integration may involve modifying source code
  - potential vendor dependencies (APIs, mgt. of security, …
  - security management involves multiple hosts and pieces of software
  - WS-Security processing is extremely hungry for CPU cycles!

Available Security Solutions - 3 -

WS-Security Gateways:

- **SOAP Security Proxy**
  - provides a virtual service endpoint (hides URL of Parlay-X Gateway)
  - messages sent to the security proxy,
    - inspected there ("content inspection"),
    - content filtering, parameter filtering, SLA enforcement
    - (if approved by the content inspection) forwarded to the Parlay-X Gateway
    - rule-based selection of *specific* Parlay-X GW (load-balancing, SLA, security)
  - **SOAP Firewall**

- **Application-level security (3-4A) gateway**
  - CPU-intensive processing offloaded to highly specialized system:
    - **authentication** (at the perimeter)
    - **authorization**
    - **confidentiality** (selective encryption/decryption)
    - **integrity** (selective XML digital signatures)
    - **audit**
    - (centralized) **administration** (of security policies, audit trails, OAM)

Available Security Solutions - 4 -

WS-Security Gateways:

- **Advantages of SOAP Security Proxies**
  - provide very comprehensive set of WS-Security standards
  - emerging security standards tightly tracked and timely implemented
  - transparent integration with heterogeneous portfolio of Parlay-X applications and other telco Web Services
    - with planned systems
    - into *existing systems* in production environments! (no coding!)
  - complete separation of application form security functions
  - CPU-intensive processing offloaded to proxies
  - often built to enterprise-grade requirements
    - performance, scalability
    - manageability
    - centralized control via remote policy server
    - high availability/ failover support

Available Security Solutions - 5 -

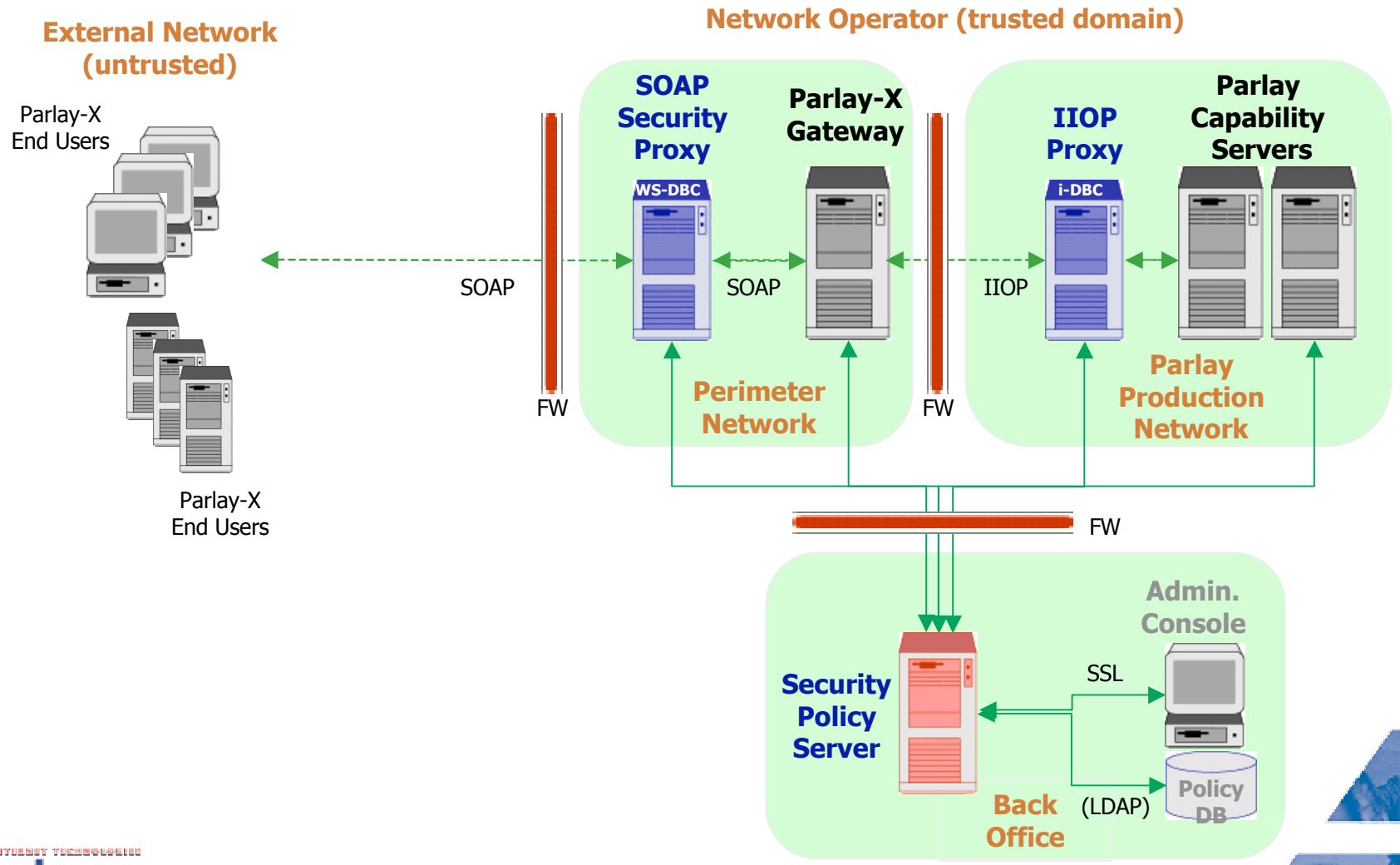
WS-Security Gateways - appliances vs software GWs:

- **appliances**
  - self-contained
  - simple installation/deployment
  - 'wire-line' speed??

- **software-based GW products**
  - flexible deployment (distribution of individual components)
    - proxies, policy-engines (on separate networks!)
    - dedicated machines (bastion hosts) or co-located with application host
    - re-use of standard h/w-backup process
  - speed!
    - Intel CPU line expected to outperform any other h/w at application-layer processing!
      - XML-Security IS a application-layer processing!
    - multi-way machines (e.g., E10K for ultimate scalability!)
  - implementation flexibility
    - timely implementation of evolving standards
    - adaptation to telco/customer-specific requirements possible

Example Solution Scenario

WS-Security Gateways protecting Parlay-X Applications



Summary: Best Practices for Perimeter WS-Security

Push SOAP/XML awareness into the network:

- **Install SOAP/XML firewalls**
  - use different OS for SOAP firewall
- **Validate all SOAP messages**
  - using XML digital signatures
- **Filter all SOAP/XML messages**
  - based on content, size, origin, authorization
- **Protect against SOAP/XML DoS**
  - block or handle PI, Entity expansions
  - validate against XML Schema to detect malicious or malformed XML
  - enforce msg size limits
  - off-load processing intensive computing task from Web services application server
- **Mask internal Web services resources**
  - use *"virtual URLs"* pointing to SOAP firewall (all internal IP addresses and URLs are hidden)
  - use URL rewrites when exporting WSDL documents

Q & A