Preview of

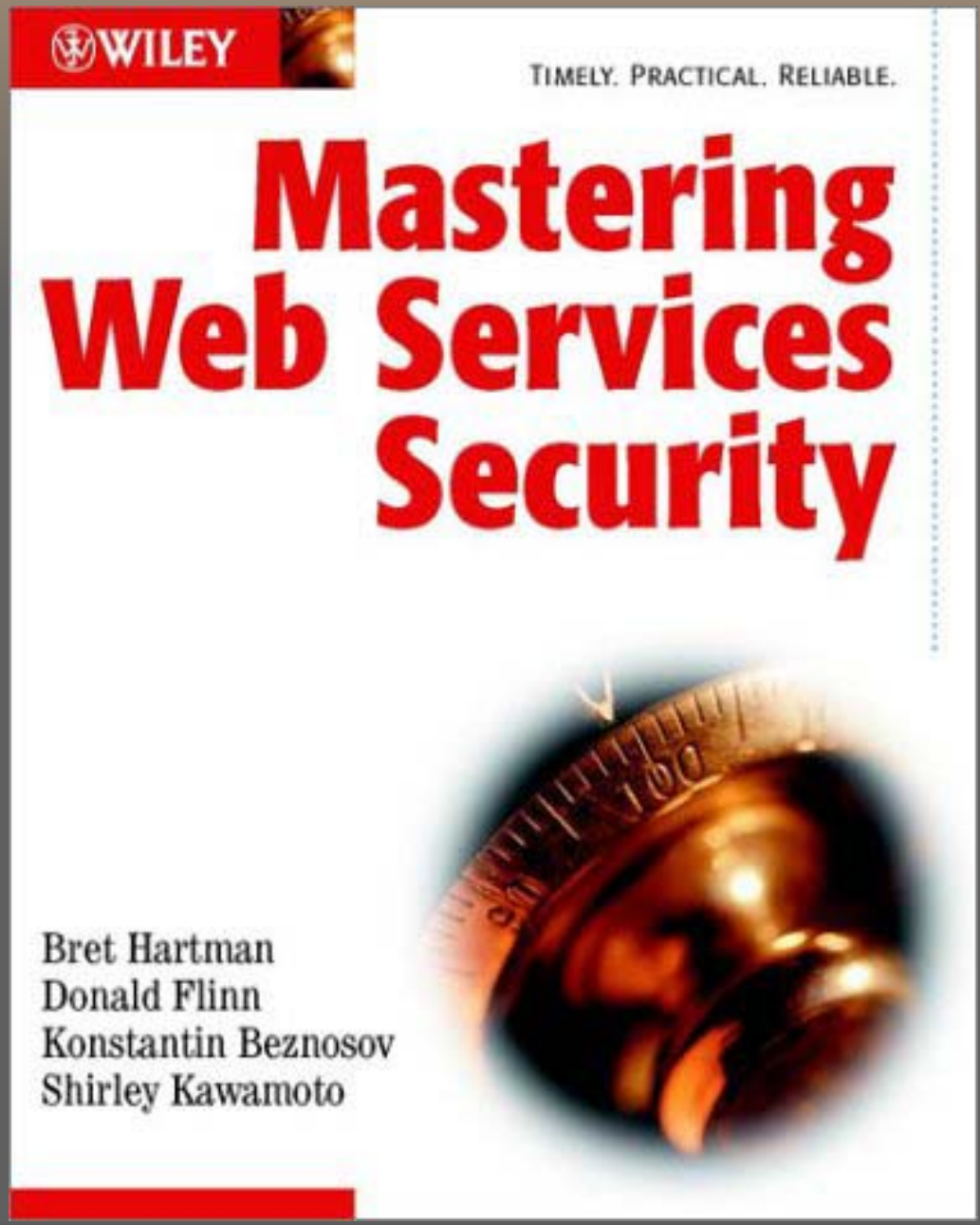
Mastering Web Services Security

Konstantin Beznosov
konstantin.beznosov@quadrasis.com
Quadrãsis

September 16, 2002

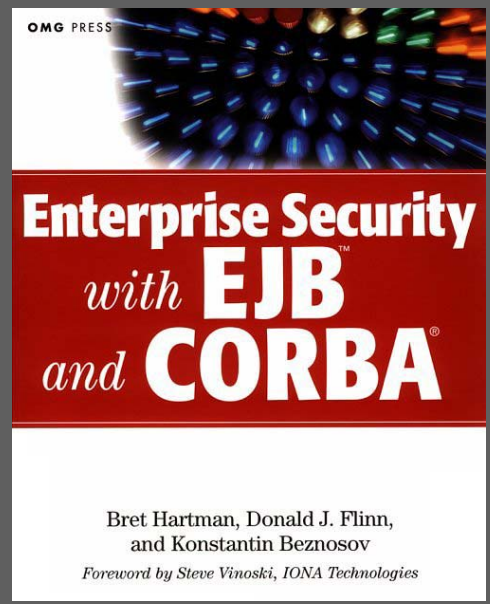
Talk Outline

- Book introduction
- Highlights of the book
  - Web Services security problem
  - XML Security
  - WS-Security
  - Security mechanisms for ASP.NET Web Services
  - Planning and building secure Web Service systems
    - Architectural and policy principles
  - EASI Framework
  - Example


WILEY
TIMELY. PRACTICAL. RELIABLE.
Mastering Web Services Security
Bret Hartman
Donald Flinn
Konstantin Beznosov
Shirley Kawamoto

Why am I talking to you?

- Ph.D. “Engineering Access Control for Distributed Enterprise Applications”
- CORBA Security
  - CORBA Security
  - “Resource Access Decision” (RAD) Facility
  - “Security Domain Membership Management”
- Security Architect
  - with Baptist Health, Concept 5, Quadrasis (HICAM)
  - Architecture, design and implementation of enterprise security solutions and products using CORBA, EJB, COM+, .NET



Book Goals

Audience

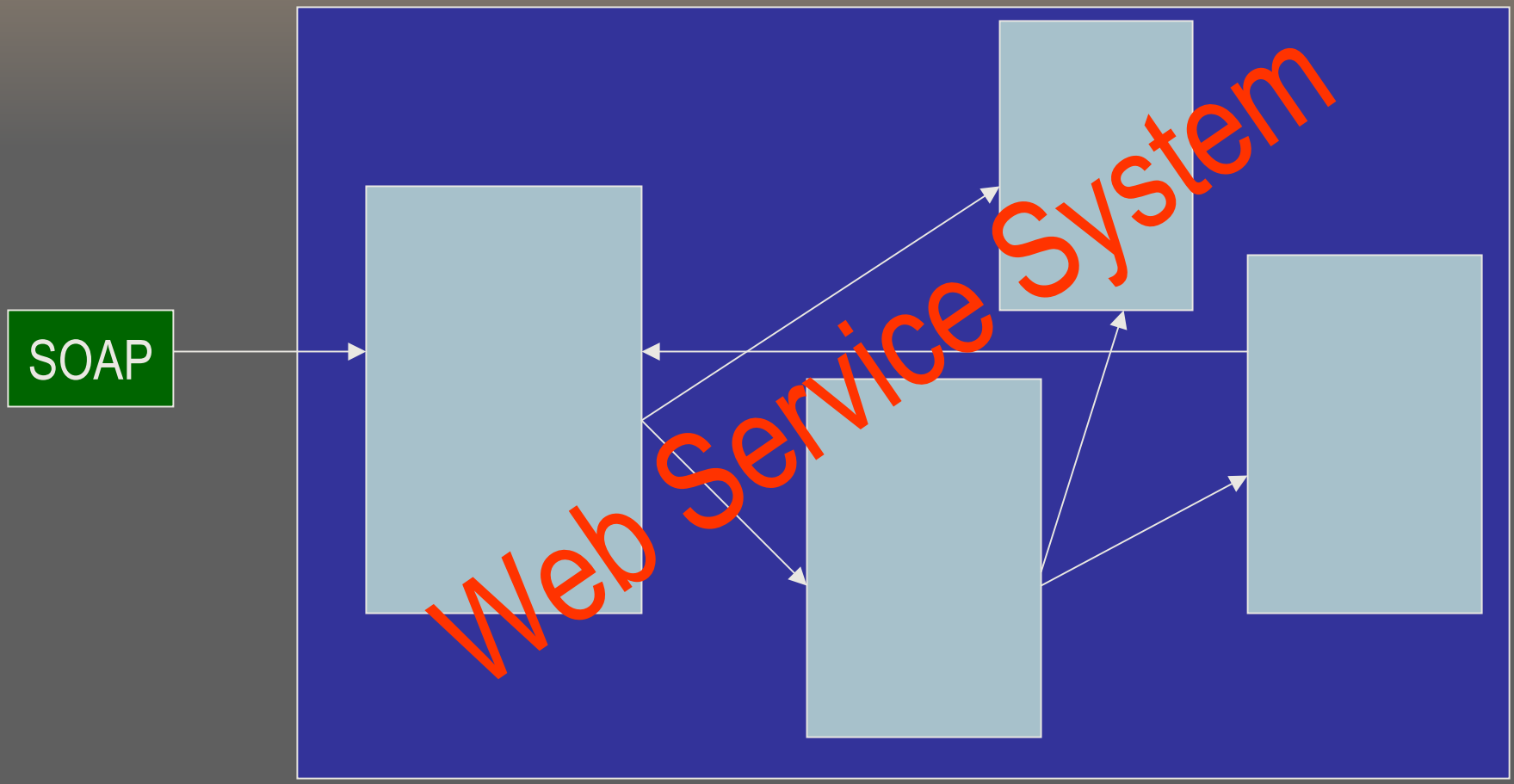
practicing application/enterprise security designers and architects

- Explain
  - key underlying principles for securing WS
  - how to secure today
    - simple WS systems
      - Java and (ASP).NET
    - complex WS systems
      - for large enterprises
- Describe what's coming and what to expect

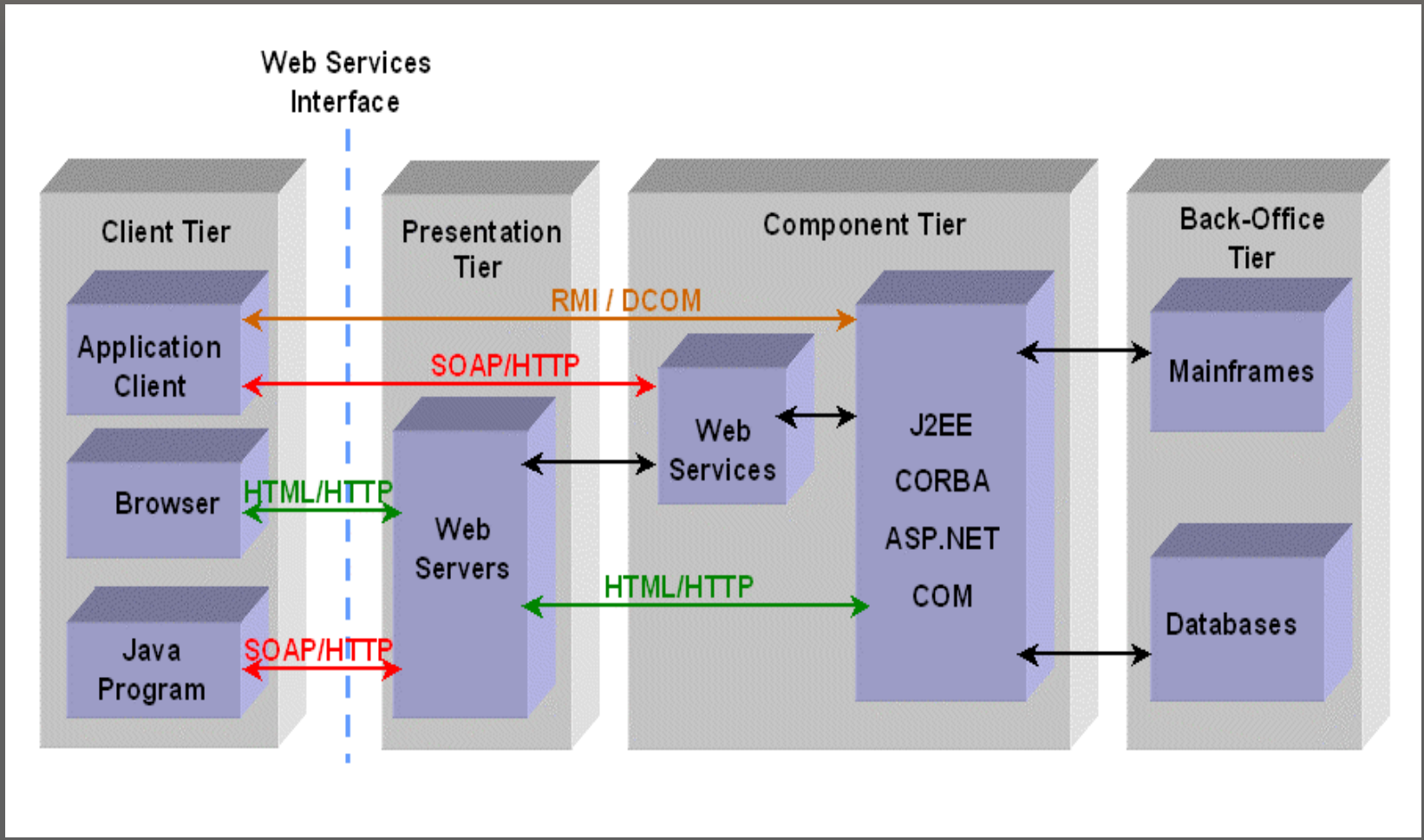
It's about

1. Principles of Securing Web Services
   - Getting Started with Web Services Security
   - XML Security
   - WS-Security
   - SAML
   - Principles of Securing Web Services
2. Middleware Mechanisms for securing Web Services
   - Middleware security mechanisms
     - CORBA, COM+, .NET, EJB
   - Securing (ASP).NET and Java Web Services
3. Advanced Topics
   - Interoperability
   - Administration
   - Planning and Building

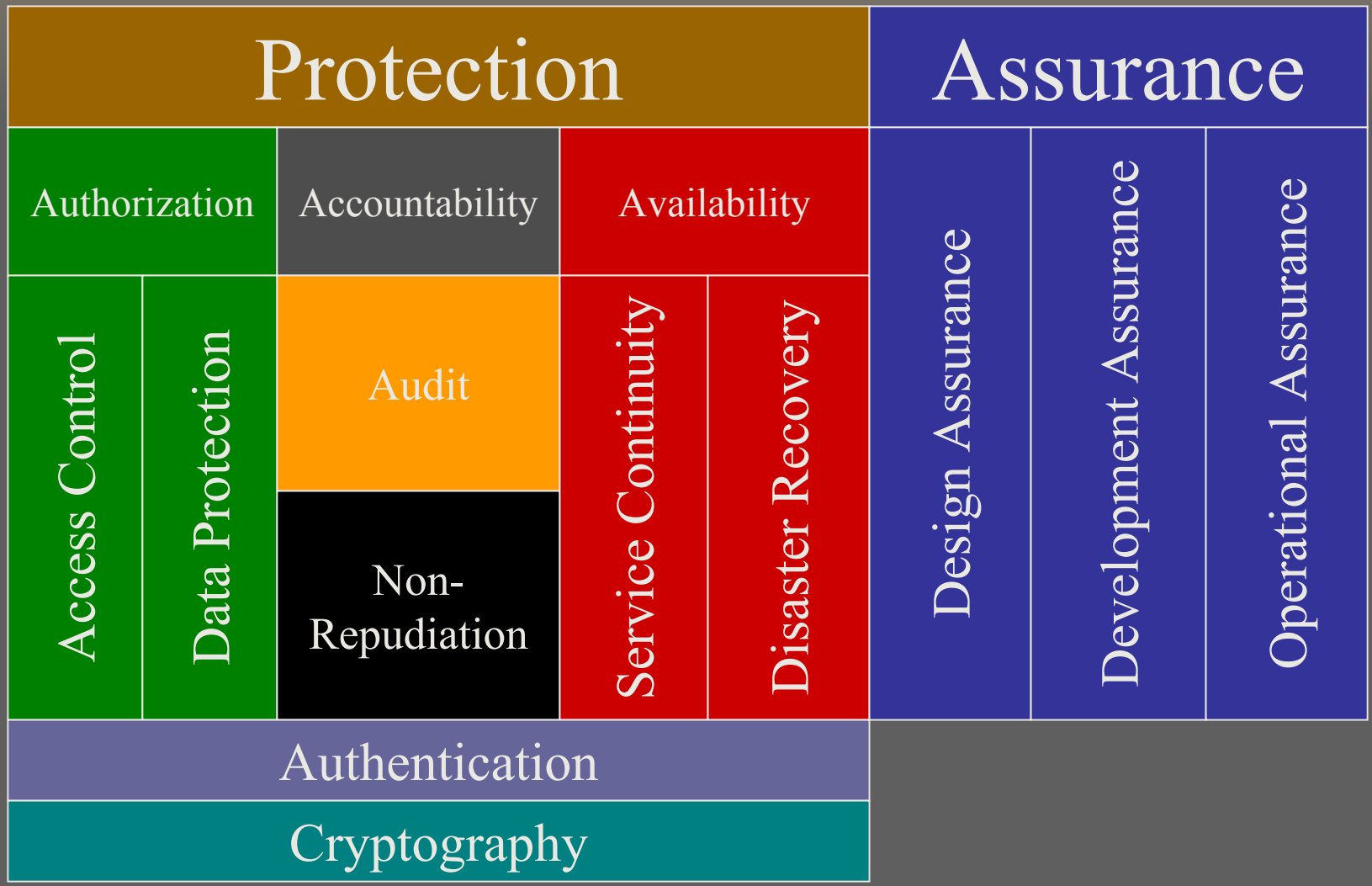
What's a Web Service System?



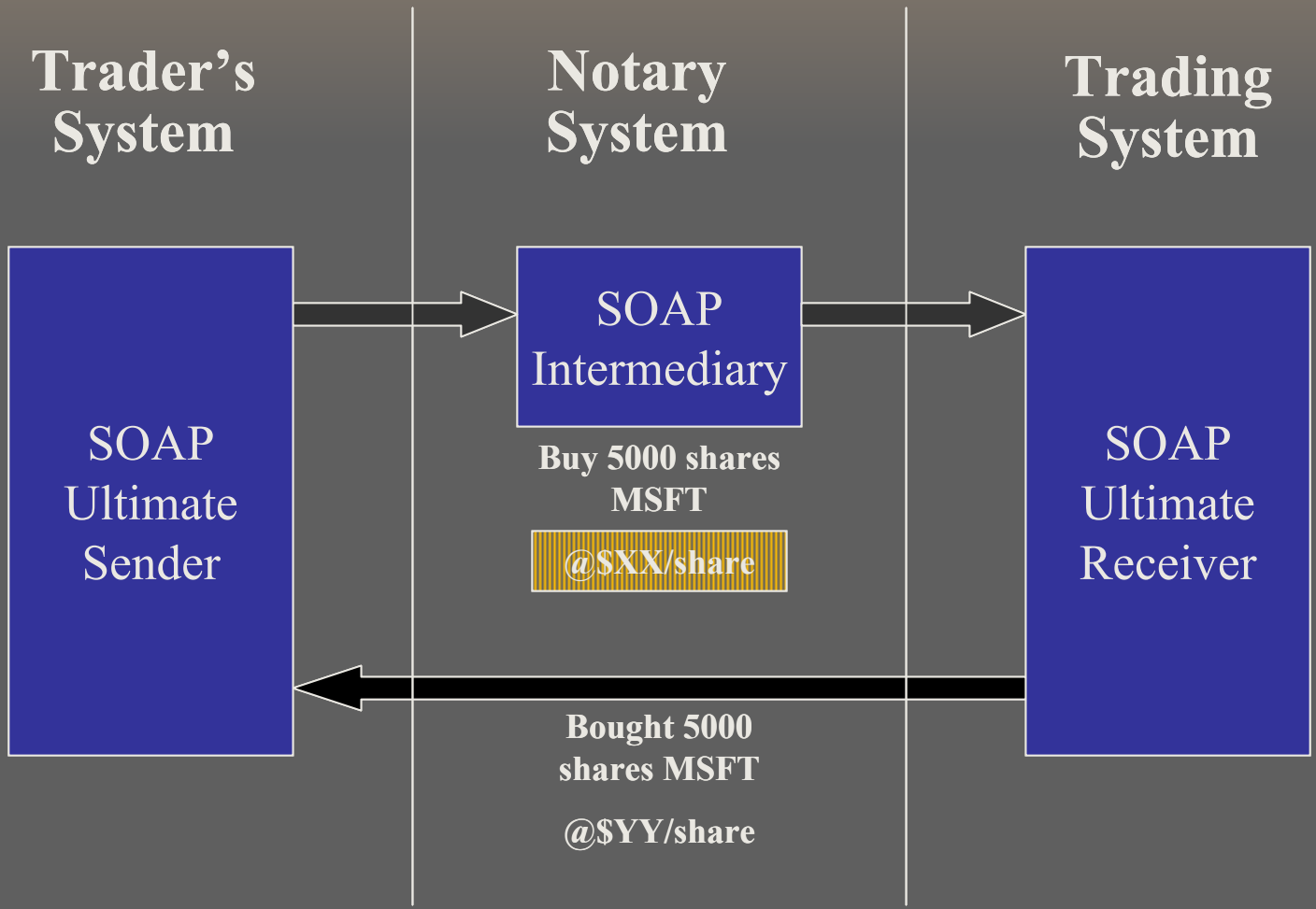
Typical Web Service Environment



Conventional Approach to Security



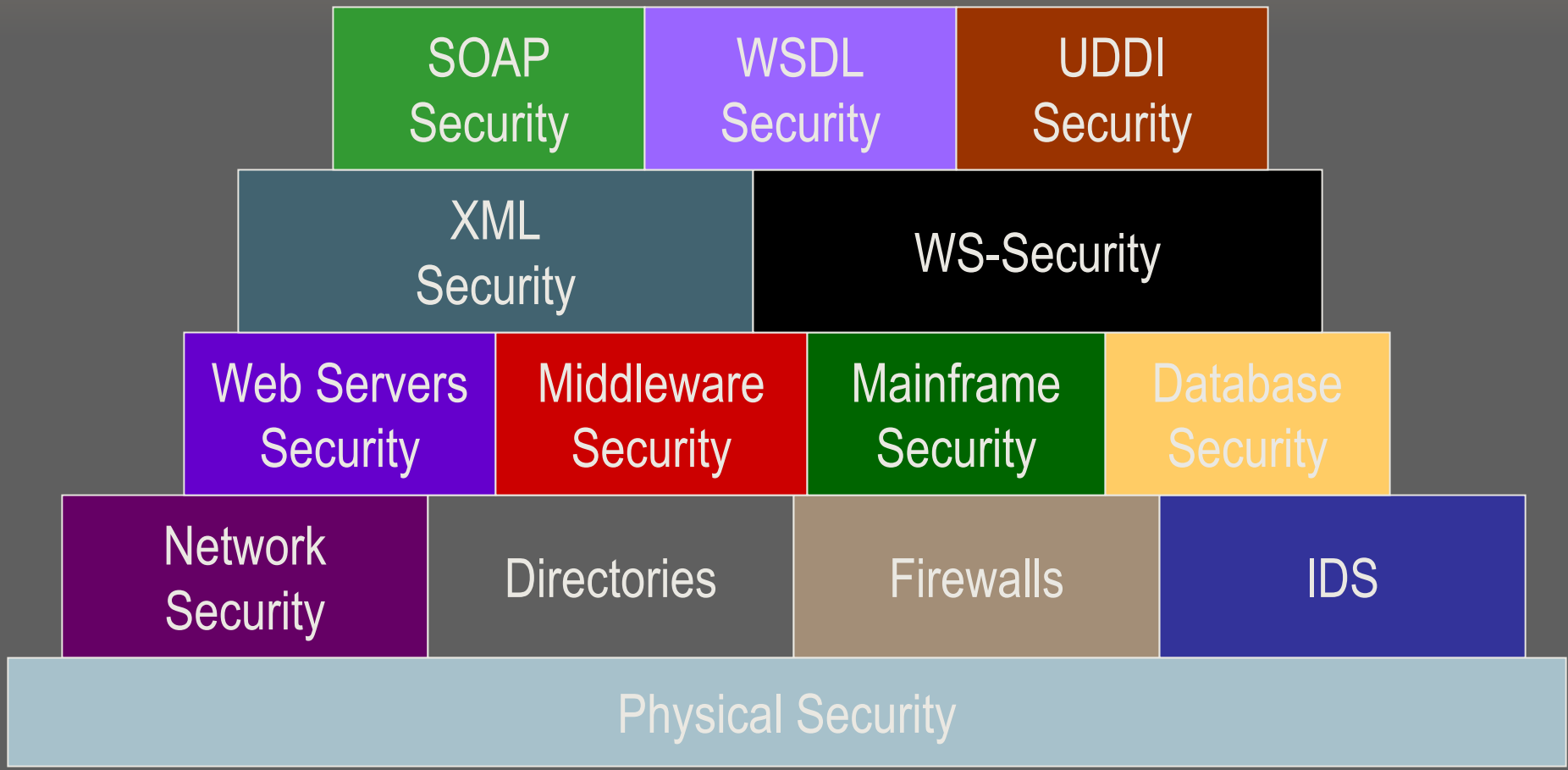
Web Usage Scenario - Security



Changes in the Security Picture

- WS open enterprise resources to outside world
- New security responsibilities due to mixing lines of business:
  - Outsourcing credit card authorization service
  - Cross-selling and customer relationship management
  - Supply chain-management
- Risk must be assessed and managed across a collection of organizations
- Interactions are more complex and take place among diverse environments

WS Security Building Blocks



XML Security

XML Encryption

- Encrypt all or part of an XML message
- Separation of encryption information from encrypted data
- Super-encryption of data

```
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
  Type='http://www.w3.org/2001/04/xmlenc#Content'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#3des-cbc'/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
  <ds:KeyName>John Smith</ds:KeyName>
  </ds:KeyInfo>
  <CipherData>
    <CipherValue>A23B45C56</CipherValue>
  </CipherData>
</EncryptedData>
```

XML Signature

- Apply to all or part of a document
- Contains: references to signed portions, canonicalization algorithm, hashing and signing algorithm Ids, public key of the signer.
- Multiple signatures with different characteristics over the same content

```
<Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Gaps

- Signature and Encryption specifications are for XML not SOAP
  - Format and location of security information in SOAP message
  - Support for multiple security operations
  - Targeting specific actors
- Passing security-related client information
  - Authentication
  - Attributes

SOAP Message Security

WS-Security

- Message integrity and message confidentiality
- Compliance with XML Signature and XML Encryption
- Encoding for binary security tokens
  - Set of related claims (assertions) about a subject
  - X.509 certificates
  - Kerberos tickets
  - Encrypted keys

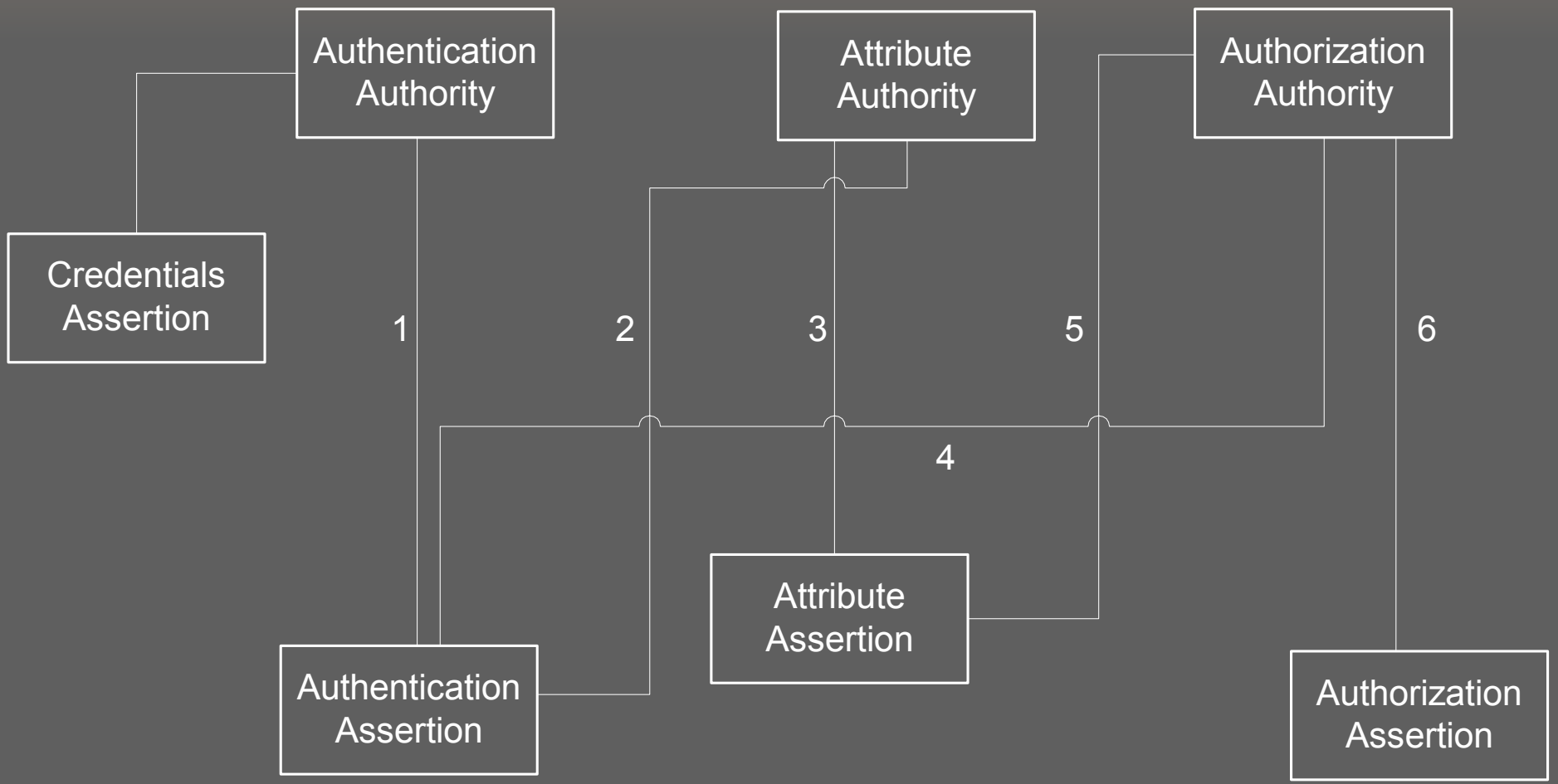
SOAP Message with WS-Security

```
<? Xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope"
  xmlns:sec="http://schmas.xmlsoap.org/ws/2002/04/secext"
  xmlns:sig="http://www.w3.org/2000/09/xmldsig#"
  xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
  <env:Header>
   <sec:Security
     sec:actor="http://www.w3.org/2001/12/soap-envelope/actor/next"
     sec:mustUnderstand="true">
     <sig:Signature>
      …
     </sig:Signature>
     <enc:EncryptedKey>
      …
     </enc:EncryptedKey>
     <sec:BinarySecurityToken
      …
     </sec:BinarySecurityToken
   </sec:Security>
  </env:Header>
  <env:Body>
   <enc:EncryptedData>
     …
   </enc:EncryptedData>
  </env:Body>
</env:Envelope>
```

Web Services Security Roadmap

- *Security in a Web Services World: A proposed Architecture and Roadmap* – April 2002
- Joint IBM and Microsoft White Paper
- Initial specifications:
  - **WS-Security**
  - WS-Trust
  - WS-Policy
  - WS-Privacy
- Follow-On Specifications:
  - WS-SecureConversation
  - WS-Federation
  - WS-Authorization

Security Assertion Markup Language (SAML)



Comprehensive Message Security

Secured SOAP Message

```
<SOAP-ENV:Envelope>
    <SOAP-ENV:Header>
      <WS-Security>
        <SAML Token>
        </SAML Token>
      </WS-Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

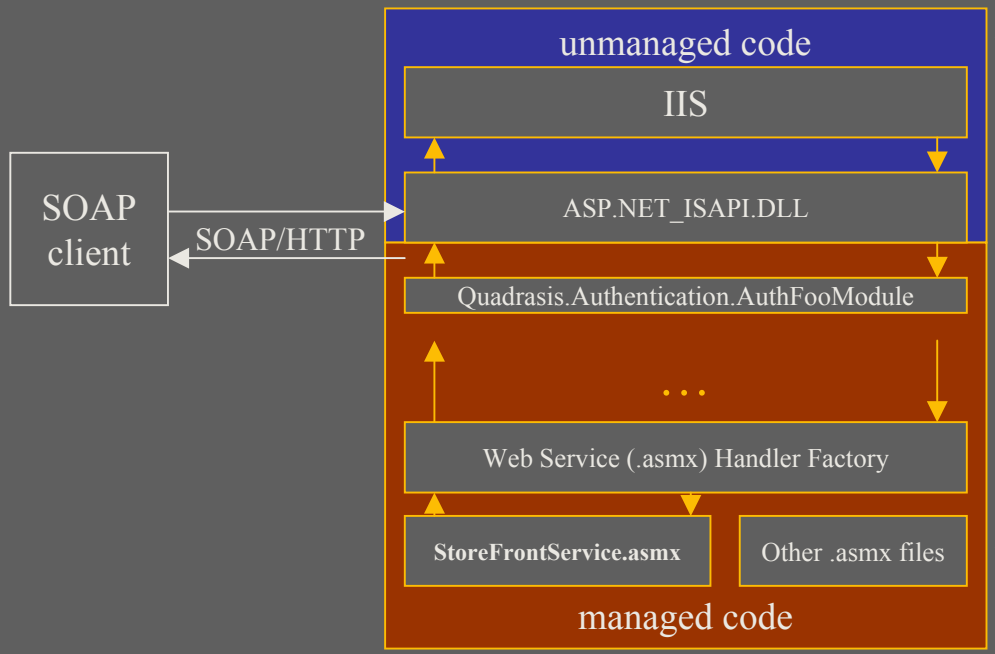
Security Feature	Function
SOAP Header	
WS-Security	- Attaches signature, encryption, security tokens to SOAP messages
SAML Token	- Authenticates initiator of SOAP request - Enables role based authorization - Time-limited - Interoperable
XML Signature, DSIG	- Multiple signed areas of header and body - Integrity protection via PKI based cryptography - Prevents tampering
X.509 Certificate (or other security token)	- Encryption and signature verification
SOAP Body	
XML Encryption	- Multiple encrypted areas of body - Prevents disclosure
RPC Method Authorization	- Prevents unauthorized call to methods
SOAP Message	
XML Schema Verification	- Validates against XML schema
Audit	- End-to-end tracing, Method access

Security Mechanisms for (ASP).NET Web Services

Options for Building MS WS

1. Publish COM+ component as SOAP Endpoint
   - Only Windows.NET and XP Pro
   - Limitations on what COM+ components could be published
   - Might be not 100% interoperable with other SOAP implementations
2. Use CLR remoting over SOAP/HTTP
   - Supports (non-interoperable) passing object references
   - Supports client and server-activated objects
   - Can be hosted by IIS
   - Vague on client authentication and channel protection, unless IIS security is used
3. Generate COM Wrapper
   - Good way to reuse existing COM components
   - No support for custom types
   - No .NET framework in the picture
4. **Use ASP.NET Mechanisms**
   - Claimed to be interoperable with other SOAP-compliant web services
   - Leverages .NET, ASP.NET and IIS security mechanisms
   - Simplifies handling of WS-Security data via WSDK

ASP.NET Custom HTTP Modules



- Advantages
  - Allows custom authentication schemes
  - Allows decoupling (HTTP) transport from SOAP
  - Makes application security-unaware
  - Supports CLR authorization
- Disadvantages
  - Couples client and server

Planning, Building Secure Web Service Systems

Recommended Approach

- Consistent with TCB principles
- Simplifies the analysis

- Leave security to experts
  - Security COTS integration vs. do-it-yourself
    - More thoroughly tested by other customers
    - More careful about common development mistakes
- Follow good architectural and policy design principles
- Plan for evolution and manageability
  - Have a security framework

Security Architecture Principles

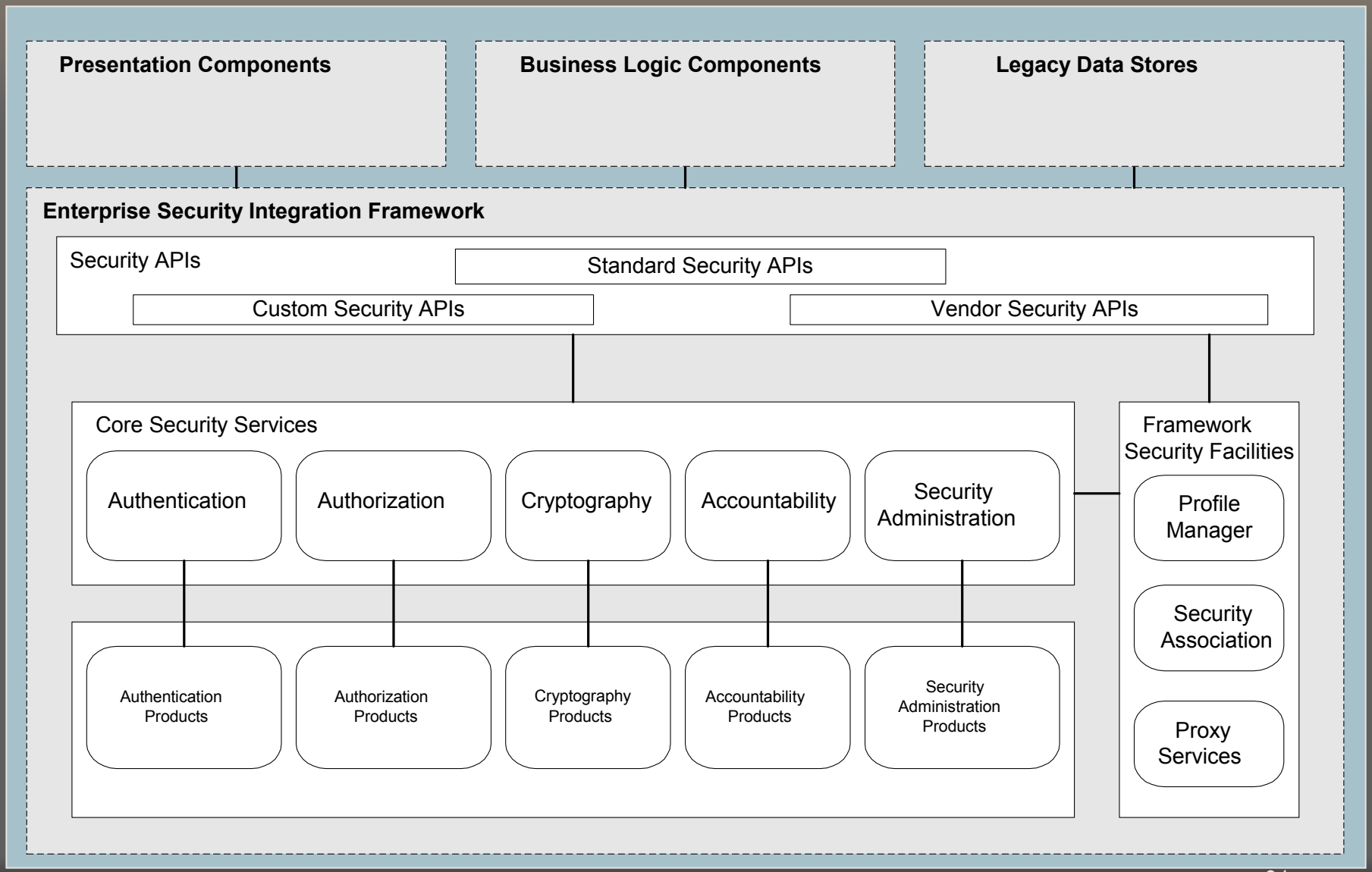
- Trust no one
  - Don't' make your firewall the only point of enforcement
  - View Web Services collections as mutually suspicious islands
- Enable interoperabiltiy
  - Use vendor-neutral standards (WS-Security, SAML)
- Modularize security
  - "Push" security down – security unaware applications
  - Insulate applications from security functionality with stable APIs

Security Policy Principles

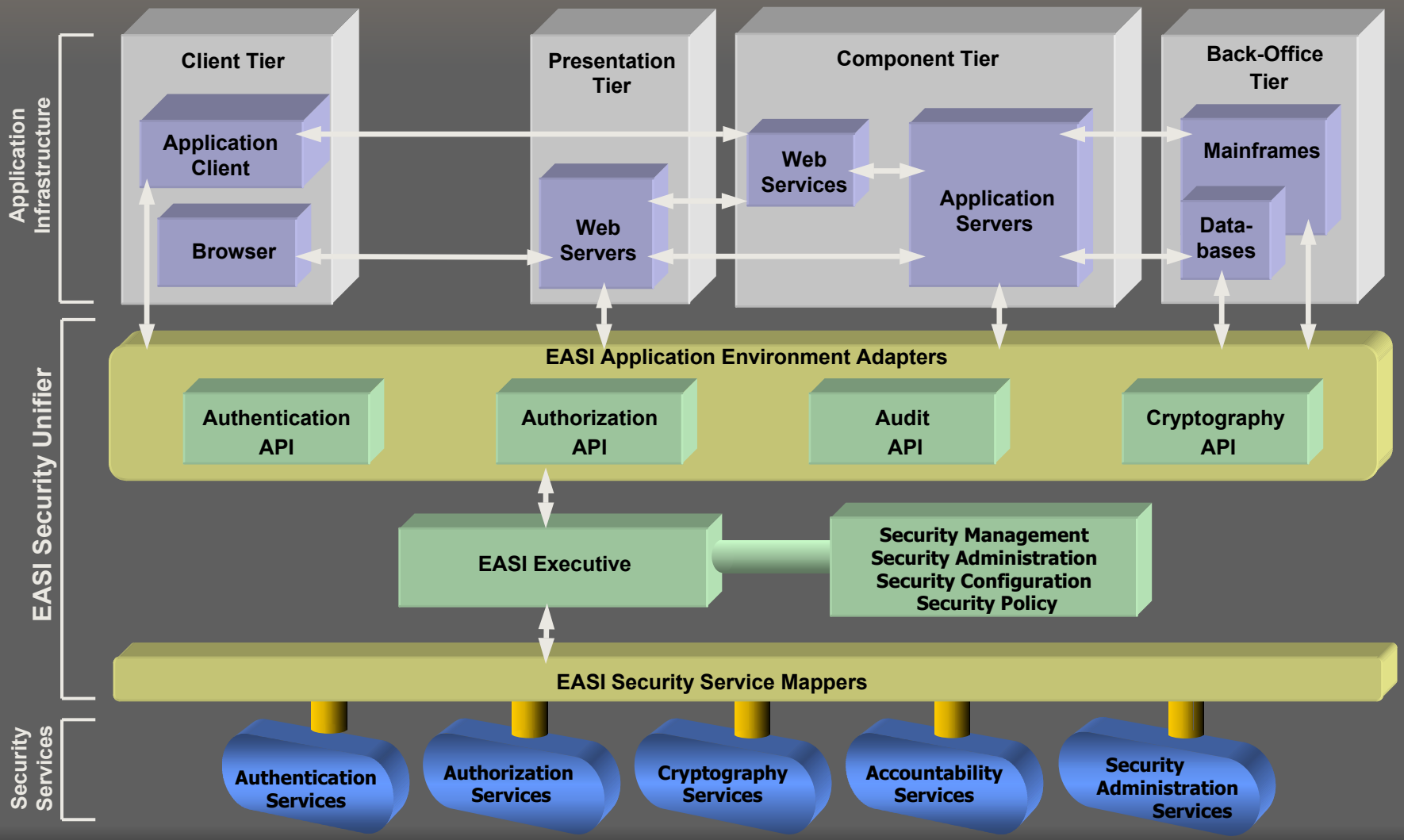
- Authentication: balance cost against threat
  - SSO
- Authorization: application-driven
  - Use the business of the application to drive authorization settings
- Accountability: audit early, not often
  - “pop” audit into/near the application
- Security administration: collections and hierarchies for scale

Enterprise Application Security Integration (EASI) Framework

EASI Framework Architecture



Specific Example of EASI: Quadrasis



EASI Pros and Cons

- Common security infrastructure shared across the enterprise
- Decoupling applications from products
- Well defined boundary between business and security logic
- No need to implement everything at once

- Complex due to generality
- Performance and scalability constraints
- Significant initial effort in designing and building it
- Has to be politically accepted in many different “parties” of organization
- Semantic mismatch among security products makes their “swapping” hard

Example

ePortal.com eBusiness.com

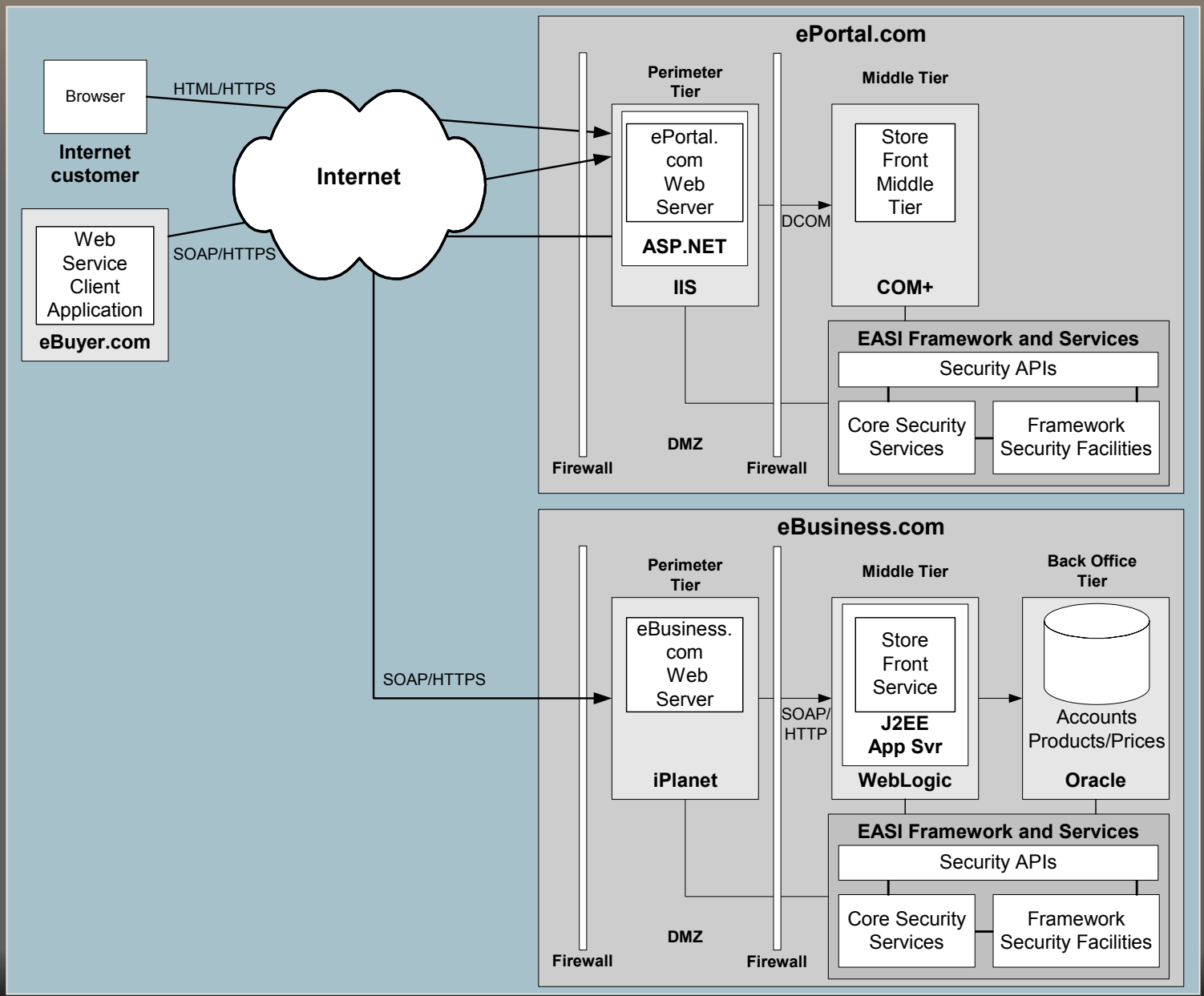
Functional Security Requirements

ePortal.com

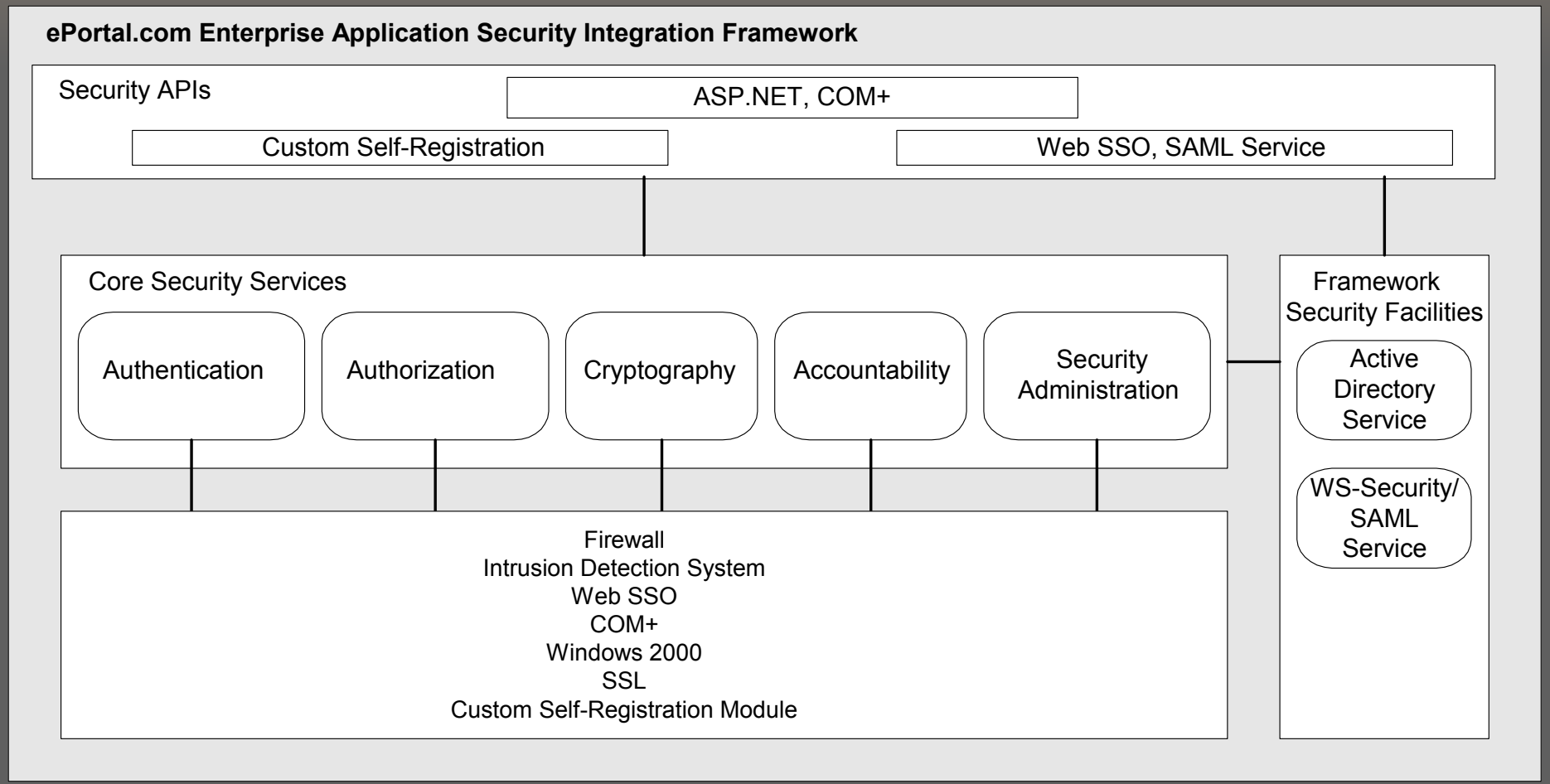
- Limit visitor access
- Eliminate administration of new customers
- Grant members more access
- Secure exchange with eBusiness.com

eBusiness.com

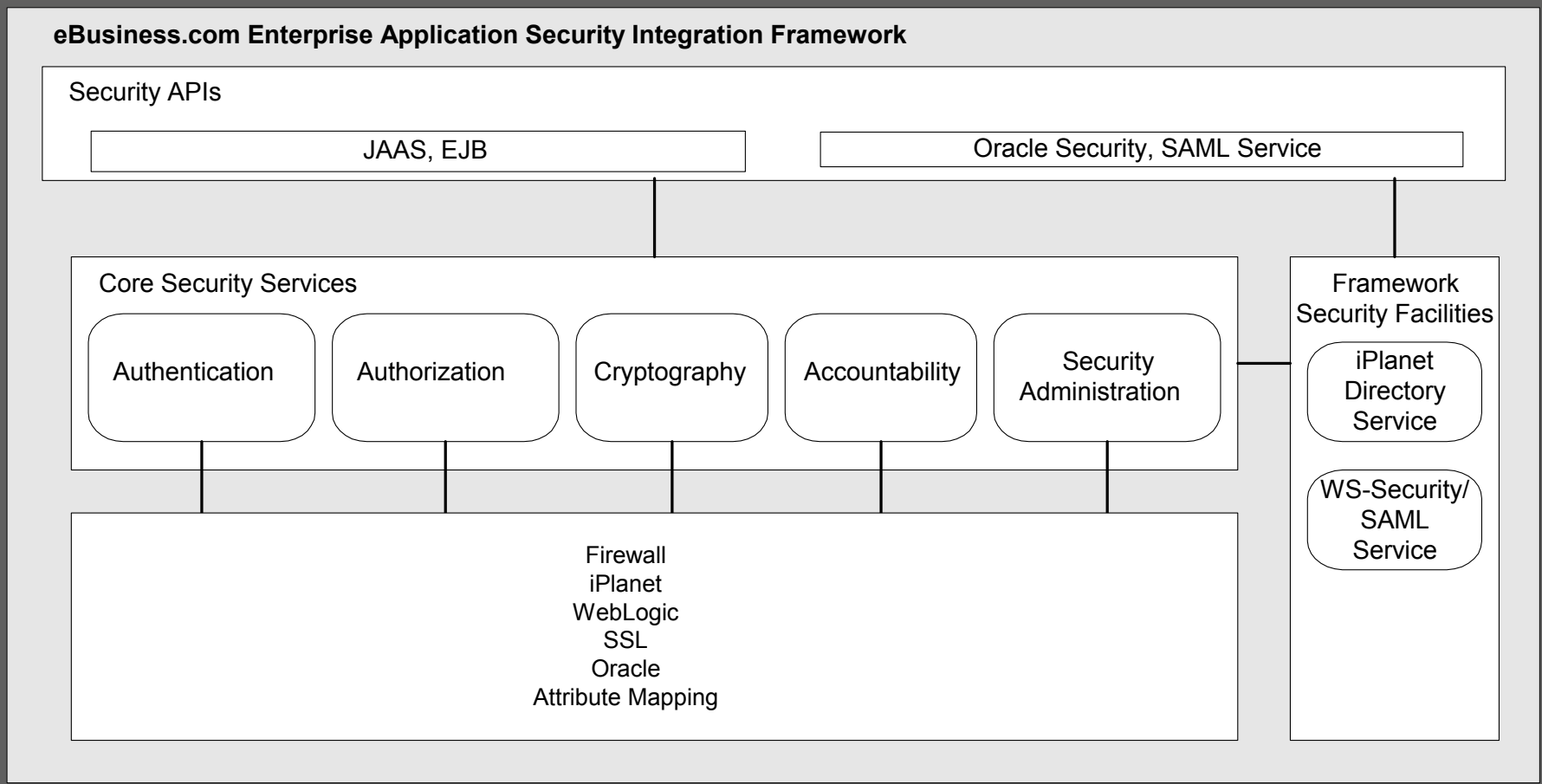
- Limit visitor access
- Protect the accounts of each individual
- Grant members more access
- Secure exchange with ePortal.com
- Administrator control of critical functions
- Restrict administrators' abilities


ePortal.com
Browser
HTML/HTTPS
Internet customer
Internet
Web Service Client Application
eBuyer.com
SOAP/HTTPS
Perimeter Tier
ePortal. com Web Server
ASP.NET
IIS
DCOM
Middle Tier
Store Front Middle Tier
COM+
EASI Framework and Services
Security APIs
Core Security Services
Framework Security Facilities
Firewall
DMZ
Firewall
eBusiness.com
SOAP/HTTPS
Perimeter Tier
eBusiness. com Web Server
iPlanet
SOAP/ HTTP
Middle Tier
Store Front Service
J2EE App Svr
WebLogic
Back Office Tier
Accounts Products/Prices
Oracle
EASI Framework and Services
Security APIs
Core Security Services
Framework Security Facilities
Firewall
DMZ
Firewall

EASI Framework for ePortal.com



EASI Framework for eBusiness.com



Security Gotchas at the System Architecture Level

- Scaling
  - Distribute requests over multiple security policy servers
  - Central administration
  - Administration delegation
- Performance – “No free lunch”
  - Encryption algorithms
  - Underlying transport
  - Policy granularity
  - Caching

Presentation Slides

- http://www.beznosov.net/konstantin