Dr Andy Hopper

"Multimedia and Network Computing"

References:


"Smart Personalisation"

References:


ORL - The Olivetti & Oracle Research Laboratory
24a Trumpington Street
Cambridge CB2 1QA
ATM Everywhere (2000?)

Wide-area switch

Σ 100G

Archive server

622M

Workstation

Storage server

High-end switch

Σ 20G

LAN

Low-end switch

Σ 8G

155M

PC

155M

Direct store

Σ 1G

25M

Module

Σ 1G

25M

Home switch

ITV

2M

Home appliances

Cellular radio

Office radio

Home radio

Mobile

ORACLE

University of Cambridge
Computer Laboratory

Copyright © 1996 ORL
2nd-Generation Applications

- Broadcast: *media server*
- Security: *look all*
- Storage: *video mail*
- Interactive communications: *video phone*

Smart feature set for all apps
# Data Analysis

<table>
<thead>
<tr>
<th>Easy</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring</td>
<td>Classifying</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Audio detection</th>
<th>Speech or music?</th>
<th>Speech recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion detection</td>
<td>How many people?</td>
<td>Gesture recognition</td>
</tr>
</tbody>
</table>
Multimedia Data Retrieval

VMR User Interface

Video
Mail
Browser

Retrieve Engine

List of
Term Indexes

Lattice Scanner

on-line

off-line

New Mail
Message Archive

Phone Recogniser

Phone Lattices

http://www.orl.co.uk/

ORACLE
University of Cambridge
Computer Laboratory

Copyright © 1996 ORL
The Ideal Network Computer

- Simple no-permanent-state terminal
- Centralised services
- All media types
- Business and consumer use
- A catalyst for ubiquitous personalisation
Teleporting

Unmodified X Clients → Teleporting Proxy Server

Teleport Session can appear on any display, or nowhere.

Teleport Controller

X Client
Mobile Streams

video processing object

video sink object

network computer

workstation

storage

multicast stream

stream mobility

user mobility

The CORBA Software Bus

STREAM MANAGER

University of Cambridge
Computer Laboratory
Beyond ATM?

Wide Area Network

Collapsed LAN (CLAN?)

- Fibre point-to-point
- Fixed-time payload
- Dedicated protocols
The pressures for change

- Inexpensive computing and telecommunication
- Increased customer expectations
- Rapid organizational change
- Globalization
- Political, economic, social, and technological

What's the real business challenge?

In the next two lectures:

- Review distributed computing technology
- Examine a real application
- Explain distributed computing

Andrew Herber (http://www.ac.co.uk)
General issues for distributed systems

- Interoperability
- Scalability
- Interconnectivity
- Can the system be deployed in small and large configurations?
- Can the system be expanded as needed?
- Can the system work with other systems?
- Can the system be deployed anywhere in the world?
- Can the system be made secure?
- Can the system be made reliable?

Models for distributed computing

- Programmability can be useful to enable functions to migrate from one computer to another
- The need for standards for interoperability
- Different parts of the system can evolve independently
- Languages interconnected by backbones
- Use networks to interconnected systems
- Provide new services to meet changing business needs
- Build on existing functional systems
- The key is evolution not revolution

Architecture

- Looking at systems in high level terms
- Design principles
- Framework of components
- Interface specifications
- Assembly rules
- Hi-level services
- High access
- User interface
- High storage
- System architecture
- Model access
Inherent features of distributed systems

Applications of distributed systems

A Real Distributed Application: Scottish HYDRO

- These differences are fundamental
- Decentralization: no single point of control
- Scalability: low cost of computing per machine
- Legacy: evolution and interfacing of existing systems
- Diversity: many types of machines in the same system
- Separation: physical and logical dispersion

Distributed systems technology

- Distributed OLE from Microsoft
- DCE from the Open Software Foundation (OSF)
- CORBA from the Object Management Group (OMG)


- and many more
- Telecommunications (network switching, control, network management)
- Command and control (military, emergency services, traffic control)
- Banking
- Real-time, on-site
- Aircraft reservations
- Diverse business areas
Remote Procedure Call (RPC)

Local procedure call can be transformed into a remote procedure call.

Abstraction #1: Remote Procedure Call in Distributed Systems

Explain the importance of understanding RPC execution semantics.

Explain the function of remote procedure call (RPC) in distributed systems.

In this segment:

- mask complexity by delegating control
- using programming abstractions
- the distributed systems technology should make network issues transparent
- keep distributed systems code separate from application code
- these inherent features make distribution complex for the application programmer
If it's worth considering whether RPC could be transparent...

...and there are also different representations for floating-point and other types...

...McGraw-VAX 11

Digital PDP-11

Intel 8086

CPU

Ordering

different data representations must be allowed for

different data representations may be represented differently (by order)

For example, integers may be represented differently (byte-ordering)

Different data representations

Types must allow for the fact that client and server may be machines of different

This is called marshalling (A→C) / unmarshalling (B→D)

and the call results

Stubs in RPC

The difference is how long you may have to wait

...just as in a local procedure call

Remote Procedure calls are normally synchronous...

Caller Waits for Callee
numbers in message

In practice, acknowledgments will often be delayed, and replication is required to handle this sequence.

Trivial implementation

Sometimes described as best effort

The error occurs before the message is acknowledged, and the peer must check to see if the server did the last operation on the message.

Inconsistent RPC semantics are appropriate for non-replicated operations.

At-most-once RPC semantics

At-least-once RPC semantics

Possible

Design for distribution - assume remote procedure call is as much as

There is no way of avoiding this issue. The conclusion is:

- Interpreted remote procedure call cannot be made highly transparent

- Ultimately, it is impossible to hide failures

In a remote procedure call you must be able to handle this

Failure of client and server

In an ordinary local procedure call you need not be concerned about independent

Remote Procedure Call Isn't Local Procedure Call
Abstraction #3: Object Request Brokers

**Object Services**

- Interface Definition Language (IDL)
- ORB = RPC + threads + local object management
- Using RPC to distribute OO programs

Abstraction #2: Threads

RPC is a general purpose protocol - it is not a replacement for optimized protocols

RPC is the lowest level building block of a distributed system

Exactly-once RPC semantics

Error handling needs to be treated carefully

- Synchronization becomes an issue
- Local processing, local LO, other RPCs
- Use threads to overlap RPC and
- How can we avoid end-to-end delays?
- RPC is blocking - wait for the reply
Any programming language for which there is an IDL language mapping definition.

Support many programming languages (C++, Smalltalk, Ada) from one interface.

They are compiled to produce stubs.

They are sophisticated header files.

They cannot be executed.

No statements.

Contain only definitions and declarations.

**CORBA Interface Definitions**

**CorbaIDL**

- An interface definition only defines interface types.
- Characteristics, for example, quality-of-service, behavior, type.
- A complete contract would define the interface.
- Each side of the interface and the obligations by the contracts.
- Client object -> server object.
- Interface definitions are a contract between service provider and service user.

**Interface Definitions**

In this segment,

- Objeets are objects.
- The OMG "standard" is the OMG Common Object Management Architecture (COMA).
- Specification.

**The OMG's Object Management Architecture**
Example C++ Interface Mapping

A simple service in CORBA IDL

- CORBA IDL in a simple build process
- CORBA applications and interfaces

One CORBA application can use many CORBA interfaces and applications

One CORBA Interface specification can be implemented by many applications

The mapping is typical
Transfer of Object References

- It must be possible to pass a reference to B's interface between A and C

Suppose A needs to tell C to use the same interface object A is using object B

In a distributed system, we need to be able to transfer object references

CORBA Types

- CORBA Types do not have specified representations
- No more, no less
- For example, short has exactly the range: -2^15 to 2^15 - 1
- Unlike some programming languages, CORBA types have a specified set of values

CORBA DataTypes

- Support multiple results
- This is a constraint on the language mappings; most programming languages do not
- Note the use of modes in our and initial

Operations are much like function prototypes...
The Naming Service

CORBA IDL does not support pointers.

The absence of pointers

get operation a parameter b parameter: the implementation of c can now call

\[ \text{interface } a \]

\[ \text{interface } b \]

For an object reference, just use the name of the interface

Object Interferences can be used freely in CORBA IDL

not just as parameters - also in structs, enumerations, union's, arrays

use object references instead

they are more meaningful in a distributed system

The Naming Service is the simplest possible directory service

The Naming Service is a white page service

Any kind of CORBA object can be named

Given a name, it will return you an object reference
Steps in Trading - (1) Server exports a service offer to the Trader

• Trading work by matching descriptions provided by clients and servers

• The server may not even exist when the client was created

• We cannot rely on clients being able to name servers

• Trading is necessary

Naming is not enough

Trading - Basic needs

Server must find a service offer that matches the request

• If there are no offers, there may be many such offers

• Server must state what it provides

Trading works by matching descriptions provided by clients and servers

• Servers will come and go dynamically

• Clients will come and go dynamically

In the future, there will be millions of interconnected servers around the world

How can clients find servers that provide the services that they need?
Incompatible, and cannot match

* If the client requests and server offer interface types do not conform, they are

Exceptions make the rules more complicated...

but they won't be used

The interoperability support at least the operations the client requires - it can provide more.

It uses the interface type conformance concept.

How does Trading decide whether a client request matches a server offer?

Matching requests with offers - type conformance

Steps in Trading - (3)

(3) Trader returns a matching service offer to the client.

Steps in Trading - (2)

(2) Client requests a service offer from the Trader.

(4) Client uses object reference to invoke the server's operations.
In the Externakization Service

In the Persistence Object (Persistence) Service

Examine how data storage is supported in CORBA

In this segment

To determine which services are needed to support business requirements, it is necessary to understand the architecture of the system. This is called the trading service model and will be discussed in detail in the next section. The trading service is a distributed system that allows multiple trading partners to participate in the trading process. There will be many trading partners providing the trading service. Because there will be millions of servers in the world:

Trading in Large Distributed Systems

Scope criteria: control where to look for orders

Preference criteria: sort matching offers into order (name, value) pairs

Other matching criteria: are known as properties

where will the service be reached?
who owns the service?
who will be used?
Preserving state

- It is an object's responsibility to preserve its dynamic state as persistent state and to recover it after a crash or any private mechanism it chooses instead.
- The object may use the Persistent Object service for this purpose.
- The state of an object can be treated as two parts:
  - Persistent state: preserved over crashes and can be used to reconstruct the dynamic state.
  - Dynamic state: typically in memory, lost if the object crashes.

Advantages of using the Persistent Object Service

- Typical data storage mechanisms do not have object characteristics.
- Uniform interfaces, self-description, and abstraction.
- This is sometimes known as an 'impedance mismatch.'
Two Worlds

Examine OMG approach distributed database and transaction processing

- Explain the Remote Data Access (RDA) technique

In this segment

Dissension on databases in distributed systems

- They may choose to hide the complexity from the client
- Object implementations do not have to provide client control
- Which copy of persistent state is to be used
- Exactly when persistent state is saved and restored
- In particular, the client may need to control
- Clients may need to control or manage persistence of the objects they use

Client Control
the relational (SQL) model as a special (optimized) case

This is an object database model

Partial unique to match by property within a relationship

The persistence service to move objects to/from disk

Structure sets of objects using relationship objects (e.g., containers)

Label objects with properties

Represent data elements as objects (with methods)

Database and the OMG

SQL (Structured Query Language)

Example

```sql
AND
WHERE
  employee佣金 > 8000
  employees`name` = 'manager'
FROM
employee
WHERE
  employee`name` IN

SELECT
  name
  commission
```

RDA Advantages

Vendor-supported interfaces to application development tools

Flexibility for clients to define application-specific views and ad hoc queries

Remote data access (RDA) overview

- The location of the remote database is transparent
- A transaction succeeds or fails as a whole
- Statements can be grouped into transactions
- A set of records, one at a time
- A single record
- Statements can retrieve, update, delete, or insert records
- Application interfaces with database using SQL statements
The Transaction Service

- Allows queries to be nested
- Allows queries to be prepared and then stored checked later
- OO for object databases
- SQL for relational databases
- Is a general query service and framework, embracing

The Query Service

- Heavily used in O-R databases (Microsoft OLE, OMG ODP/ODC)
- Represents relationships as complex data objects
- 2 relationships 4 roles 9 objects involved
- Reference to relationships
- Links connecting documents, documents containing figures
- An example of containment

The Properties Service

- Attributes cannot
- Properties can be read and deleted
- Client applications can get and set both properties and attributes
- Adding a new attribute means changing the IDL
- Attributes defined in CORBA IDL are static
- Allows typed named values to be dynamically associated with objects
Events are concerned with asynchronous communication between objects.

The Event Service

X/Open IS0 TP Model

Typical TP Core Components (much simplified)

What's Done Must Not Be Undone

Durability

Concurrent transactions must not affect each other

Isolation

Transactions must be self-contained logical units of work

Consistency

All-or-nothing

Atomicity
Look at an example of secure electronic commerce

Introduce network security

Introduce WWW and Java

In the next two lectures.......

Events could be used to build an RM (Robust Queueing Message) Interface

so-called ‘push’ and subscribe

There can be multiple suppliers and multiple consumers

via an event channel

Suppliers and consumers of events are decoupled

for example, when disk space is getting low

“events; changing modifications

Events support asynchronous modification

Events
HTTP is a communications RFC
TCP/IP is the Transport Control Protocol/Internet Protocol
Built on top of TCP/IP, and using TCP/IP transparently
Defined by an ongoing series of Internet-drafts
HTTP is the HyperText Transfer Protocol
HTTP

What is the WWW?

THE ESSENTIALS ARE STANDARDIZED BY USE AND THE IETF

There is no center

It is truly distributed

The WWW is the largest information system ever created

What is the WWW?

A network of webmasters and browsers

For the purposes of this presentation, the WWW is

News
Mail
Gopher
Telnet

The WWW includes and simplifies access to

No narrow definition of the WWW exists. It

Please definition not possible

What effect will this will have on electronic businesses

How the WWW is affecting the development of distributed systems

Java and JavaScript

HyperText Markup Language (HTML)

HyperText Transfer Protocol (HTTP)

Describe the World Wide Web (WWW) and its essential components

In this segment
The Common Gateway Interface

Many services are based on delivery of information that must be

Web server delivers documents in response to browser requests

WWW Browsers

HTML

WWW Browsers

The user's local machine resources are used for the prestation

Communicates with the WWW server via the HTTP protocol

Responsible for retrieval and presentation of HTML documents

The browser is a client in distributed systems infrastructure

HTML 1.0 allows for access from within range of terminal types

exact presentation is under control of user's browser

HTML 1.0 is consistent meaning of presentation standard for each "chunk"

HTLM 2.0 is currently being drafted by a standards task force

HTML is the HyperText Markup Language

• HTML

HTML does not define presentation of a document rigidly

HTML 2.0 is commonly misused for presentation standards that are borrowed

Some vendors have made nonstandard extensions

Originally defined as a subset of SGML

• HTML

HTTP is the Hypertext Transfer Protocol

The HTTP process then

- handles the request and the connection to the client
- rejects the receive process to handle the request
- When it hears an incoming HTTP request
- Serer is a process that listens on a predefined port
- Web server delivers documents in response to browser requests

Web server delivers documents in response to browser requests

The child process then

- handles the request and the connection to the child
- rejects the receive process to handle the request
- When it hears an incoming HTTP request

The child process then

- handles the request and the connection to the child
- rejects the receive process to handle the request
- When it hears an incoming HTTP request
Java Language Design

Java is designed for distributed systems technology.

Java programs are "secure" in a programming sense, but not in a systems sense.

No pointer arithmetic or conversions
Java and C++ assumptions about memory space structure
Uses C++ based ideas and syntax
Java is object-oriented
Although owned and controlled by Sun
The Java language specification is public

Java Application Anatomy

Java

import java.util.Date;

class DatedApp {

    // your code here
}

public static void main(String args[])
{
    System.out.println("DatedApp program started.");
    Date today = new Date();
    public static void main(String args[])
    {
        // your code here
    }

Introduction to Java

"...and run remotely" in the user's browser

"...http://..." for the virtual machine can be moved across the WWW

If runs on a virtual machine, so is platform independent
Java is special with regard to the WWW because
"...and a few understandable features removed"
with a few improvements added"

"...basically a lot like C++"
Java is a normal programming language

and checked for security issues
Since these can easily be distributed and shared
The WWW community favors scripts (or "perl" case Java)
They can be written in any programming language
They must be extremely well tested before installation
The server can only run from itscgi-bin directory
They are specially privileged
CGI Programs are pieces of code runable by the Web Server
WWW Security

Remote services through secure programs on Web server
Web services must guard against security compromises

Java Security

• Bytecodes executed inside an applet interpreter
• Bytecodes authenticated using public key encryption techniques (see later)
• Bytecodes verified for safety during compilation
• Language restricted to disallow memory accesses

Java is intended for networked distributed environments

Remote Method Invocation (RMI) and Serialization

• Together give mobile objects
• Can be sent as method arguments and results
• Inherit from Serialization classes to represent state and byte codes
• Much better integration than CORBA++
• References marshaled as remote references (C. CORBA)
• Objects marshaled as data
• Type libraries are not required
• Inherit from Remote objects to pick up RMI libraries and find (marshaling) methods

Java Virtual Machine and Bytecodes

• Bytecodes include information about interfaces
• Source code is not distributed
• First verified by compiler
• Only Bytecodes are shipped from Web server to browser
• No implementation dependencies provided
• Bytecode (mostly of byte sequences)
• The interpreter is architecture neutral
• This virtual machine is the Java bytecode interpreter
• Java compiler generates Bytecodes for a virtual machine
Network Security

- PGP: Pretty Good Privacy
- Simon Garfinkel, O'Reilly & Associates
- Know how to configure PGP
- Understand basic concepts of cryptography
- Understand security issues in network systems

Even More Information

- More information on WWW security
- More information on HTTPS
- More information on FTP
- More information on confusion in the field of security

Firewall Defenses

- Firewalls are the standard defense for web servers
- Two-tier firewall configurations are favored
- Web servers is placed inside other firewall
- Corporate Internet is behind an inside firewall
- Proceded with a huge array of defenses
- Proceeded by packet filters

SeeFirewalls and Internet Security by Cusick & Bellow (Addison-Wesley 1994),
For more on firewalls
For more on WWW security
For more on PGP security
For more on encryption protocols. Algorithms and source code in C. Bruce Schneier.

**Private Key (symmetric) Cryptography**

- International Data Encryption Algorithm (IDEA) - 128 bit used in PGP
- RC2, RC4 - 1 to 1024 bit - first round RIPEML / RSA Security
- Triple-DES (DES three times with two keys) - used by banks
- Data Encryption Standard (DES) - 56 bit
- Problem of key distribution - how to send a single common key which both keep secret
- Both sender and receiver share a single common key which both keep secret

**Keys**

- Different encryption is of similar plaintext
- chosen plain text (cool yet no sending a message of your choosing)
- known plain text (e.g. mailing list, consumer number, etc.)
- K1^ - K2^ - age of the message = 1.2^ - 10^ (1)
- brute force - try every possible key (128 bit key at 10^9 trials/sec on 10^9 computers =
- breaking the code
- Key escrow conspiracy may be the way out...
- Years many governments resist key export
- The longer the key, the more secure the encryption system delivers
- Some many government resist import or export of algorithms
- Keys are just numbers

**CONCEPT CHART**

- How can you stop people from eavesdropping your messages?
- Non-repudiation
- How can you stop other people reading your messages?
- Confidentiality
- How can you prove a message hasn't been altered?
- Integrity
- How can you prove who you are to a distant host in a way which can't be mimicked by an impostor?
- Authentication

**Encryption Algorithm**

- Decrypt
- Cipher text
- Plain text
- Key

**Cryptography**

- The message after is encrypted (cipher text)
- Message you wish to encrypt (plain text)
What encryption can't do

Can hide the fact a message was sent (traffic analysis)
Can protect you against a tailored
Can protect against a buffer-overflow exploit program
Can protect against denial-of-service (denial of service)
Can protect against stolen keys
Can protect your unencoded information

What encryption can't do

Use RSA to establish symmetric session key
RSA is slow

Benefits:

Encryption and decryption algorithm are the same

Rivest-Shamir-Adleman (RSA) Algorithm

sometimes called asymmetric cryptography

A message encrypted with one can be decrypted with the other (either way)

Secret key must be physically secure
Public key can be used to encrypt (based on a dictionary, a newspaper, etc.)
Public key can be used to encrypt (based on a key, generated as a pair)

Public Key Cryptography

works best for many-to-one communication
KDC has to be online all potential users, all the time
KDC is obviously prone to attack (physically)

Disadvantages:

But beware "sympathetic" and "man-in-the-middle" attacks
KDC can do mutual authentication as part of key distribution
Any recipient can ask the KDC to issue a short-term secret session key
Member may need to ask the KDC
Use a secure key distribution center
don't scale

Need a different key for each pair of communicating parties

Private Key Distribution
Applications of digital signature

Server returns a challenge for the server name - e.g. www.xyz.com

User verifies session key and sends it encrypted in server's public key

Digitally sign email messages or HTML forms to validate others

Server returns a digital signature, i.e. a challenge

User authenticates digital signature and sends a secret key encrypted with server's public key

Do you trust the author of the message?

Digitally sign Java applets or other downloaded code

Open PGP signed email by typing “pgp -e” and entering the password of

Accept validity of keys signed by 5 or more people you trust

PGP model users exchange signed keys to build up web of trust

Certificate authority publishes signed public key to user name mappings

Certificate authority publishes signed public key to user name mappings

Certificate authority publishes signed public key to user name mappings

Same as digital signature

Digital signature protects integrity of message

Table of contents: England

Signatures integrity of whole message

Examples include public keys and certificates with transmitted data

Digest of a message

A one-way function which discards the message info into a large (e.g. 128-bit) number

Compute a message digest (e.g. MD5 or SHA algorithm)

To sign a message

It is possible for some agencies to trust my bank’s my government’s friends’

To encrypt this with your (well-known) public key

Receipt of the ciphertext and your secret key

Encrypt the ciphertext with your secret key

But who made the digest?

To sign a message

Digest of a message

A one-way function which discards the message info into a large (e.g. 128-bit) number

Compute a message digest (e.g. MD5 or SHA algorithm)

To sign a message

It is possible for some agencies to trust my bank’s my government’s friends’

To encrypt this with your (well-known) public key

Receipt of the ciphertext and your secret key

Encrypt the ciphertext with your secret key

But who made the digest?
Electronic Commerce

- Cross-border transactions
- Risk management -失眠, controllability
- Cryptographic policy

- The big issues
- Low cost electronic data interchange
- Business-to-business
- n't it shopping, but successful
- Allow me to mediate, telephone order
- Customer-to-business
- Share Internet users worldwide and growing
- Using the Internet to buy and sell

Smartcards

- Can reach device requires a PII of biometric data to be entered
- Self-destructs if tampered with (old the bad guys are getting clever...)
- Stores keys and does copy protection
- Convenient, easy to use and read
- Card can be sized computer with a simple CPU and memory
- Use a smartcard
- Encrypt them with a pass phrase?
- Keeping keys on your desk isn't safe
- Keys are too complex to memorize

Security and Transparency

- Easy to add to Java
- Some P2P (e.g. BitTorrent) builds into Java
- Integrity
- Confidentiality
- Authentication and access control
- Can do:
  - Hand to do non-repudiation at a transparency, except at message level
  - Protocols are application-specific

Non-repudiation

- All most security attacks are on operational processes rather than cryptosystems
- Designing security protocols is hard
  - Sequence numbers and timestamps and known Planck's / different parameters
  - How do you stop synchronized clocks?
  - Include sequence numbers and time stamps in messages to avoid replay attacks
  - Protect the entire path to avoid to replay
- In complex exchanges sign each message and include signed digest of previous
  - Message exchanges, sign each message and include signed digest of previous
Security Issues

- SSL for confidentiality
- Web server is scalable
- No need for web front-end for public key infrastructure
- Access is always "on-line"
- Transaction is many to one
- User population is small
- Symmetric key distribution since

The Secure Electronic Transactions Protocol

- Get payment information from gateway
- Make purchase (client), merchant, payment gateway
- Get authorization (client), merchant, payment gateway

Steps

1.外墙 - 1和数据传输协议的加密以阻止"窃取"和"欺诈"攻击
2.分隔交易数据包含的有关，如需查看细节
3.在交易信息的发行方，以电子支付卡为基础的支付平台
4.银行给出的智能卡

Proposal by Visa and Mastercard, based on public key cryptography.
Security Issues

- Need to sign applet (and user has to trust us)
- No cell overhead - simple end-to-end channel
- Interface can be customized to individual user's needs at runtime
- Have to trust security of customer's operating system

Alternative with Java

- HP issue browser plug-in with smartcard to perform operations
- Plug-in enforces Java containers (via smartcard crypto engine and keys)
- Business server dehydrates messages and processes transactions with access control
- Internet (green zone)
- Internet server insulated in "amber zone" between Internet ("red zone") and
- Smartcard
- Business server simply acts as a message switch
- Have to trust security of customer's operating system
- Full access control is applied in amber zone