**Network Computing Infrastructure**

**Middleware**

**Operating system**

**Network**

**Applications**

**Application components**

**Analogy**

- Auto manufacture and devises process of assembly lines
- All auto companies components, e.g., batteries
- Professional services, e.g., accounting, law
- Resource management, e.g., janitorial, gardening
- Support interaction at different locations

Those are specialized services not typically provided within the company
Two ways to design a system

Decomposition from system requirements

Assembly from available components

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Components

When the functionality of subsystem reaches its maturity, but allows for customizing

Component: A subsystem purchased “as is” from an outside vendor

(Alternative – building your own subsystem)

A component implementation is encapsulated (although often configurable)

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice.
The Linux OS we are buying “off the shelf” and integrating into our architecture. The Linux OS is a **component**.
Other Examples of components

Computer
Disk drive
Network
Network router
Operating system
Integrated circuit
Database management system

Why is a component implementation encapsulated?

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Outsourcing: A subsystem design is contracted to an outside vendor.

Responsibility is delegated.
Suppose we choose to pay another firm to develop the user interface.
This is called **Outsourcing**.
Why would we do this?
Suppose we bring together all these subsystems and test them...

This is called **System Integration**
System integration

- Bring together subsystems;
- make them work together;
- to achieve a goal.

Requires

- Testing
- Making modifications to
  - architecture and/or
  - subsystem implementation

---

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Supplier Types

Three types of suppliers:

- **Component Suppliers**
  - Specialized in one or a set of related components

- **Custom Subsystem Developers**
  - Taking customer’s requirement and meet their needs

- **System Integrators**
  - Implementing, assembling and integrating components

(E.g., Computers (microprocessor, disk, drive, etc), computer manufacture: 1 & 2)

Slide adapted from slides for *Understanding Networked Applications*
By David G Messerschmitt. Copyright 2000. See copyright notice
Two ways to sell Software

Product

Customer installed and operated
Often (but not necessarily) sold or licensed at a fixed price

Service

Functionality provided over a wide-area network
Often (but not necessarily) sold by subscription

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Recall: Infrastructure and Applications

Infrastructure
- Equipment and/or software used by many applications

Applications
- Provide specific capabilities and features serving individual users.
Four possibilities

Product | Service
---|---
Microsoft Office | Hotmail
Application
Infrastructure

Personal computer | Internet DNS

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Software Products

Two types

- **Push**
  - Suppliers develop and define a product and sell it off-shelf
    - **Example:** Microsoft Windows

- **Pull**
  - Customers provide a specification and commission a supplier to develop such application
    - **Examples:** ecommon

(Mostly a off-shelf but allows for modification!)
Application Service Provider (ASP)

- Two types
  - Bundled
    - An infrastructure provider bundles applications with their infrastructure
      - Example: Comcast, telephony service providers, Ooma?
  - Unbundled
    - A provider of an application service without providing an infrastructure service
      - Examples:
Examples of unbundled ASP model

- Yahoo: Web-based calendar
- gmail: Web-based email
- Schwab: Web-based stock trading

Slide adapted from slides for Understanding Networked Applications by David G Messerschmitt. Copyright 2000. See copyright notice.
Unbundled ASP model

Advantageous to “user”

- Proven way to reduce installation, integration, and maintenance costs
- Contractual obligation for availability and quality
- Location independence
Unbundled ASP model (con’t)

Advantages to supplier

- Ongoing revenue stream supporting upgrade and maintenance
- Usage-based revenue better aligned with user’s value proposition
- Opportunity for price discrimination (e.g., standard, premium), advertising revenue, etc.

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Some pricing alternatives

Price discrimination?
Usage dependent?
Terms and conditions
- fixed, leasing, per-use, subscription
- warrantee, service level agreements

Bundles
- maintenance, support, releases, provisioning and operations

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Infrastructure acquisition

Infrastructure

- Build and operate
- Build but do not operate
- Do not build but operate
- Neither

Trend

Outsourced operations
System integrator
Service provider

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Application acquisition

Application

Develop internally

Buy as product

Contract development

Product w/ customization

Trend

Software supplier

Outsource developer

Supplier, consultants
Stovepipe vs. Integrated Infrastructure

**stovepipe architecture**
---or---

**Turnkey Solution**

- Single supplier provides all encompassing solution
- (complete with infrastructure)

**Integrated Infrastructure**

- Separate infrastructure that can support many applications

---

Slide adapted from slides for *Understanding Networked Applications*
By David G Messerschmitt. Copyright 2000. See copyright notice
From stovepipe to layering

Data | Voice | Video

Application-dependent infrastructure

Many applications

Integrated Infrastructure (Maybe broken into Additional layers.)

Application-independent
**Stovepipe vs. Integrated Infrastructure**

---

**Stovepipe Architecture**

---or---

**Turnkey Solution**

- Eg., landline telephone
- One supplier involves, takes full responsibility
- E.g., vertical integrated utilities (*others?*)

**Integrated Infrastructure**

- Supplies focus on either applications or infrastructure.
- Easy to deploy new applications
- Layered structures

---

Application and Infrastructure

Application

Infrastructure

---

Slide adapted from slides for *Understanding Networked Applications*

By David G Messerschmitt. Copyright 2000. See copyright notice
Stovepipe vs. Integrated Infrastructure

- **Economies of scope**: allow supporting a variety of applications
- **Economies of scale**: allow supporting a variety of applications, thereby lowering unit costs
- **Lower marginal cost**: Each new application leverages on existing infrastructure
- **Larger market**: owing to exiting infrastructure, suppliers see low risk with great potential
- **Diversity of applications**: resulted from low marginal cost & larger market
- **Competition**: User can mix and match complementary technologies.
Vertical Integration vs. Diversification

A company is **vertically integrated** when it makes rather than buys the subsystems in its products by acquiring suppliers that previously sold it components.

A **diversified** company produces products across different industry segments, achieve synergies, financial stability, e.g., google, costco (gas, tire), etc.

---

Slide adapted from slides for *Understanding Networked Applications*  
By David G Messerschmitt. Copyright 2000. See copyright notice
Vertical Integration vs. Diversification

- Why do customers favor less vertical integration?
  - Prefer competition amongst component suppliers
  - Mix and match components
  - Reduced lock in

- Disadvantages??
  - Customer needs to integrate components from different suppliers.
Vertical Integration vs. Diversification

Why do customers favor diversification?

- Reduce internal coordination costs by having to deal with fewer suppliers or simply face a single service provider.

(because suppliers also produce products across different segments! Think about what can you do with google-related services, including google doc, calendar, photo, etc or Apple-related ones.)

Slide adapted from slides for Understanding Networked Applications By David G Messerschmitt. Copyright 2000. See copyright notice
General Trend

- Less Vertical Integration
- More Diversification

- Internal coordination cost down
- External coordination cost down
- Reduce risk, product cycle
- Provide customer with complete solution
- Synergies across different product lines

- More vertical integration
- Internal coordination cost down
- External coordination cost down

- Make Diversification

- Customer demand for open system
- Competition improve quality & prices

- Make

- Buy
Today’s supplier structure

- Applications
- Frameworks and components
- Middleware
- Infrastructure (network, OS) software
- Equipment (network, computers)
- Semiconductors, components

Slide adapted from slides for *Understanding Networked Applications*
By David G Messerschmitt. Copyright 2000. See copyright notice
Purpose of a standard?

- Allow products or services from different suppliers or providers to be interoperable*

- Increasingly component interoperability cannot be dependent on end-user integration, e.g., think about that you have to put a computer where there are various non-standard interfaces existed.

*Components are interoperable when they interact properly to achieve some desired functionality
Scope of a standard*

Included: *: a specification generally agreed upon

- interfaces (physical, electrical, information)
- architecture (reference model)
  - Standard way of decomposing a system so that suppliers (competitive, complementary) can follow
- formats and protocols (FAP)
  - Define how interface works, e.g., parameters, return, etc
- compliance tests (or process)

(Establish a ongoing process of upgrading, improvement, e.g., extensions)

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Reference model

Decide decomposition of system
- where interfaces fall

Defines the boundaries of competition and ultimately industrial organization
- competition on the same side of an interface
- complementary suppliers on different sides
- hierarchical decomposition at the option of suppliers
- (possibly) optional extensions at option of suppliers
Some issues

Once a standard is set

- becomes possible source of industry lock-in; overcoming that standard requires a major advance, e.g., “biogas (E85)” others?
- may lock out some innovations,
  - e.g., might be alternative of doing thing that is not compatible with the standard
Types of standards

*de jure*
- Sanctioned and actively promoted by some organization with jurisdiction, or by government

*de facto*
- Dominant solution arising out of the market
- Voluntary industry standards body

Industry consortium
Common or best practice

Examples? “walk on the right side”, “drive on right side”
Examples

*de jure*
- GSM, ISDN Telephone interface

*de facto*
- Microsoft Windows API (Application Programming Interface)
- Intel Pentium instruction set,

*Voluntary industry standards body*
- IEEE (Institute of Electrical and Electronic Engineers)
- IETF (Internet Engineering Task Force)

*Industry consortium*
- bluray

*Best practice*
- Windowed GUI

Slide adapted from slides for *Understanding Networked Applications*
By David G Messerschmitt. Copyright 2000. See copyright notice
The changing process

- As technology and industry progress quickly, the global consensus standards activity has proven too difficult
  - e.g. ISO
- "New age" standards activities are more informal, less consensus driven, a little less political, more strategic, smaller groups
  - e.g. IETF
Programmable/extensible approaches for flexibility
  - e.g. XML, Java

Slide adapted from slides for *Understanding Networked Applications* 
By David G Messerschmitt. Copyright 2000. See copyright notice
Reasons for change

From government sanction/ownership to market forces

- Increasing fragmentation, i.e., not easy to regulate
- Importance of time to market, i.e., regulatory process is typically slow

Greater complexity

- Less physical/performance constraint for either hardware or software

---

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Lock-in

(Particularly open) standards reduce consumer lock-in

- Consumers can “mix” and “match” complementary products

Standard increases supplier lock-in

- Innovation limited by backward compatibility*, i.e., suppliers need to respect existing standards

(An earlier version conform and can be used in situation with new standards, e.g., MPEG.)

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Aside: Network Effects

- The value of owning some products goes up if lots of other people have it too.
  - Examples? “dropbox”, “line”, others?

- This phenomenon is called “network effects”

- How do standards influence network effects?
Network effects

Standards can harness network effects to the industry advantage

- Revenue = price x market size x market share

Increases value to customer (market size)

---

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
**Why standards?**

*de jure* are government driven to reduce confusion and cost

*de facto* standards are sometimes the result of positive feedback in network effects, not recognized by any formal body, open used by anyone, but may be with a proprietary implementation, e.g., post script.

Governments like them because they

- promote competition in some circumstances
- may believe they can be used to national advantage
Open vs. Proprietary Standards

- Open standard - a standard that is well documented, unencumbered by intellectual property rights and restrictions, and available to any vendor, i.e., through academic research

- What are the advantages? e.g., no lock-in for consumers, stimulate innovations

- What are the disadvantages? e.g., proprietary technologies may find it difficult to compete
Why companies participate

Pool expertise in collaborative design
  - e.g. MPEG

Have influence on the standard

Get technology into the standard
  - Proprietary, with expectation of royalties

Reduced time to market
Standards applied to Business Processes?

- Can you standardize business processes?

- Yes!
  - ISO 9000
    - A set of standardized business processes for Quality Management.
    - Supports TQM (Total Quality Management)
  - RosettaNet
    - A set of standardized business processes, and accompanying standardized data interfaces/formats for conducting e-business.