

A background network diagram consisting of a grid of light gray lines forming a mesh of squares. At the intersections of these lines, there are small, semi-transparent gray squares, creating a pattern that resembles a network or a molecular structure.

Peer-to-Peer Technologies for Sharing and Collaborating on the Net

Presenter: Camilo Rostoker,
B. Sc. Honours
Department of Computer Science,
University of Regina

Presentation Overview

1. Introduction to P2P Computing Concepts
2. The History and Emergence of P2P
3. P2P Architectures & Infrastructures
4. Enabling Technologies
5. Current P2P Platforms
6. Introduction to JXTA
7. Conclusion

What is Peer-to-Peer Computing?

- **Distributed Computing:**
 - in general, is any computing that involves multiple computers remote from each other that each have a role in a computation problem or information processing.
- **Peer-to-Peer Computing:**
 - Peer-to-peer is a communications model in which each party has the same capabilities and either party can initiate a communication session.

“P2P is how we do it, not what we do”

- Chris Williams, CEO of Oculus

P2P Fundamentals

- The key fundamentals of P2P computing is the idea of **sharing** and **collaboration**.

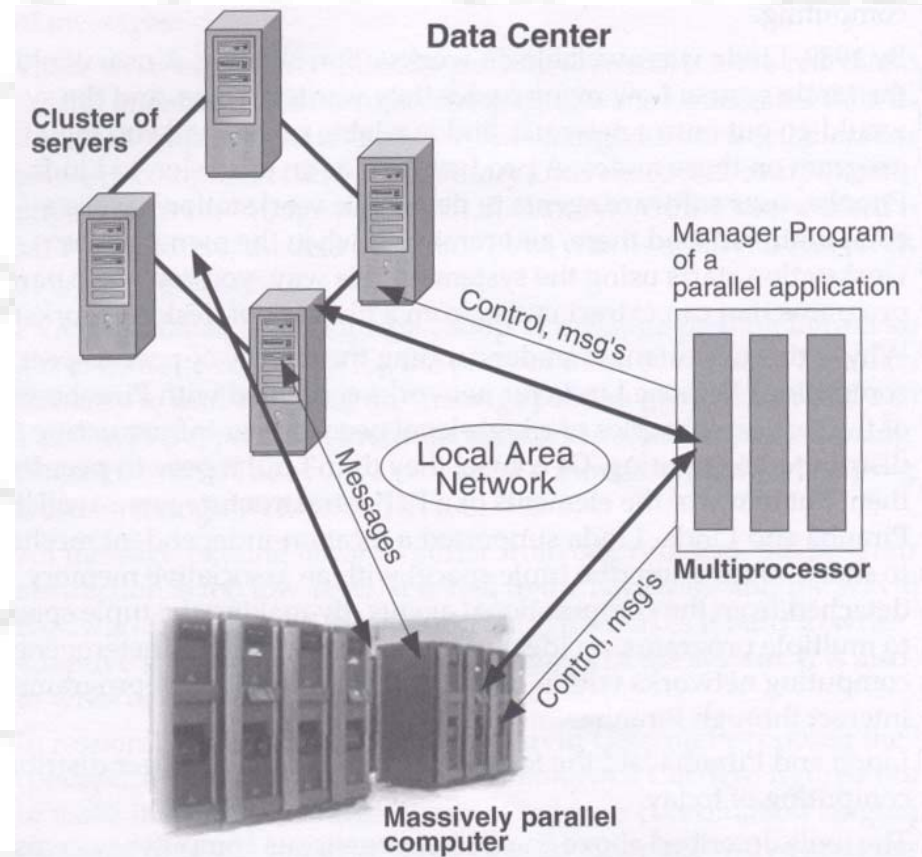
Resources that can be shared:	P2P Collaboration Techniques:
<ul style="list-style-type: none">•Files and compute-cycles•Computations•Storage space•Information•services	<ul style="list-style-type: none">•Direct exchanges between peers•Transactions via intermediary•Granting access to resources•Interactions formed through policy (e.g. online communities)

The History Behind P2P

- P2P Computing is **not** new
- New explosion fueled by enabling technologies and standardization of protocols
- ARPAnet, the predecessor of the Internet, was designed to model the real world → 1:1 direct exchanges
- Xerox PARC (Palo Alto Research Center)

Early Distributed Computing

- 1970's:
 - Massively Parallel Machines (MPP)
 - Parallel Virtual Machine (PVM)
- 1980's:
 - Linda & Piranha
- 1990's:
 - Message Passing Interface (MPI)



Distributed Computing the Old Way

Early Collaborative Applications

- Bulletin Boards Systems (BBS)
- Internet Relay Chat (IRC)
- Groupware's
 - Microsoft Exchange
 - Novell Groupwise
 - Lotus Notes

Why and Why Now?

Technical Appeal	Social Appeal
<ul style="list-style-type: none">▪ Access huge amounts of untapped resources▪ Eliminate single-source bottlenecks▪ Eliminate single point of failure▪ Enable distributed services between clients▪ Remote maintenance capabilities	<ul style="list-style-type: none">• People can create their own online communities and run them as they choose• People do not like centralized authority• Near-perfect anonymity is possible

Why and Why Now?

Why Now?

1. Internet
2. Hyperlink
3. Multiple standardized protocols: HTTP, Telnet, FTP, Gopher, etc
4. Request for Comments (RFC)
5. Graphical-Interface browser
6. Widespread Internet access
7. Broadband Internet access

Areas of P2P Development

- Virtual Communities
- E-business
- Gaming
- Search Engines
- Virus Protection
- Collaborative Development
- Large-Scale Computations

Challenges of P2P

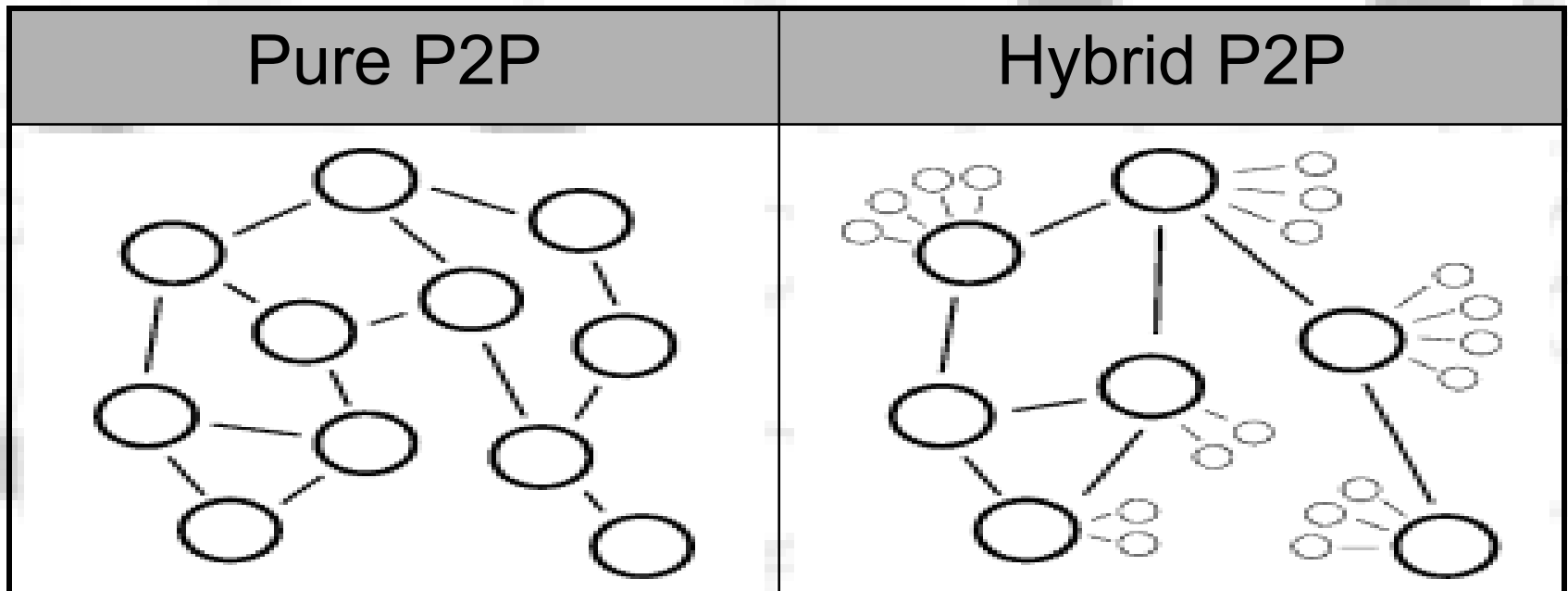
- What are some of the major challenges for the future of P2P?
 - Security and Privacy
 - Scalability
 - Connectivity & Intermittency
 - Interoperability
 - Performance and Bandwidth
 - Hardware, Software and Network Heterogeneity

P2P Architectures

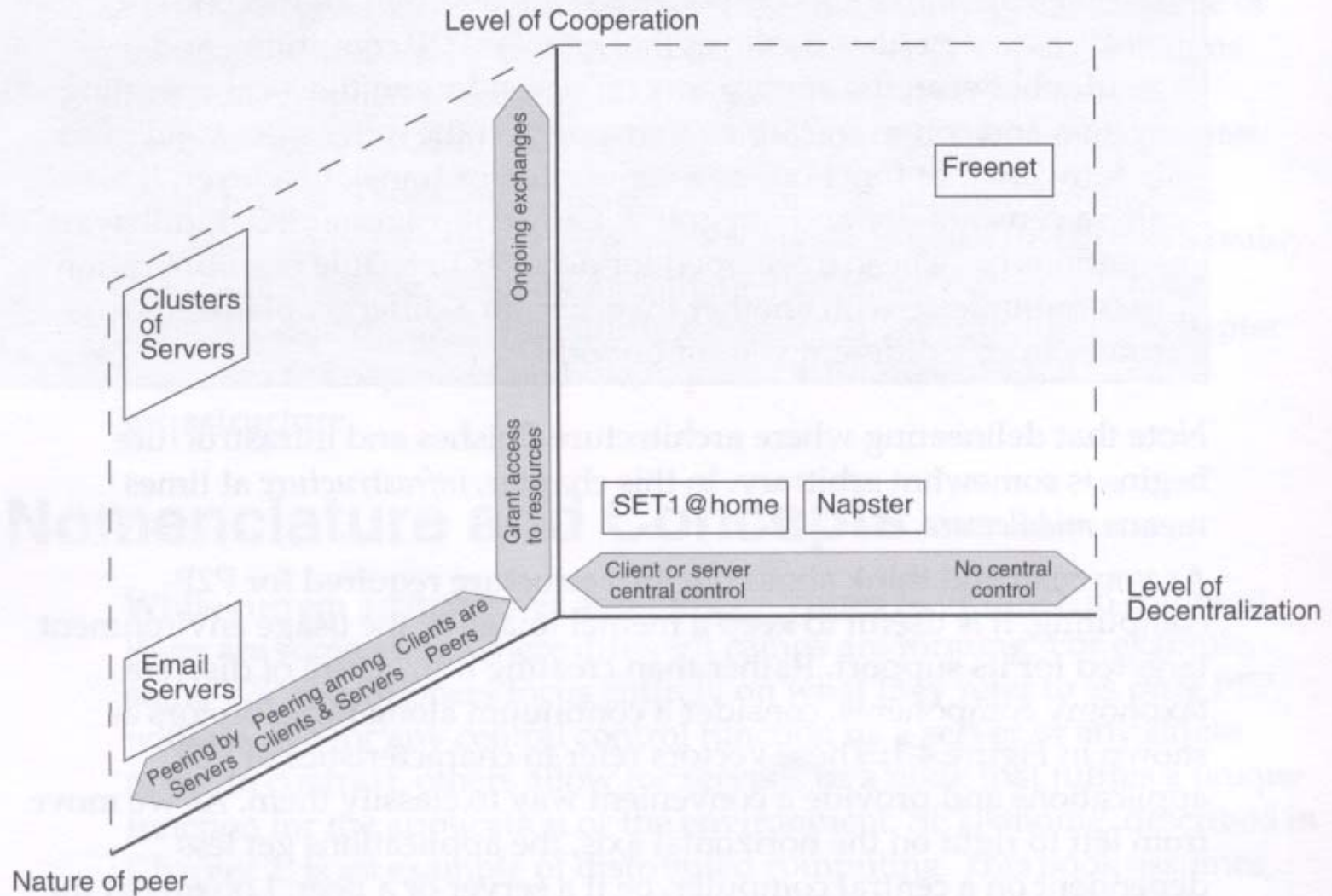
- **Architecture:** defines the way components fit together in a system
 - **Software Architecture**
 - **Network Architecture**
 - **P2P Architecture**
- **Infrastructure:** the underlying framework and building blocks

P2P Architectures

- Essentially there can be 2 main types of P2P Architectures



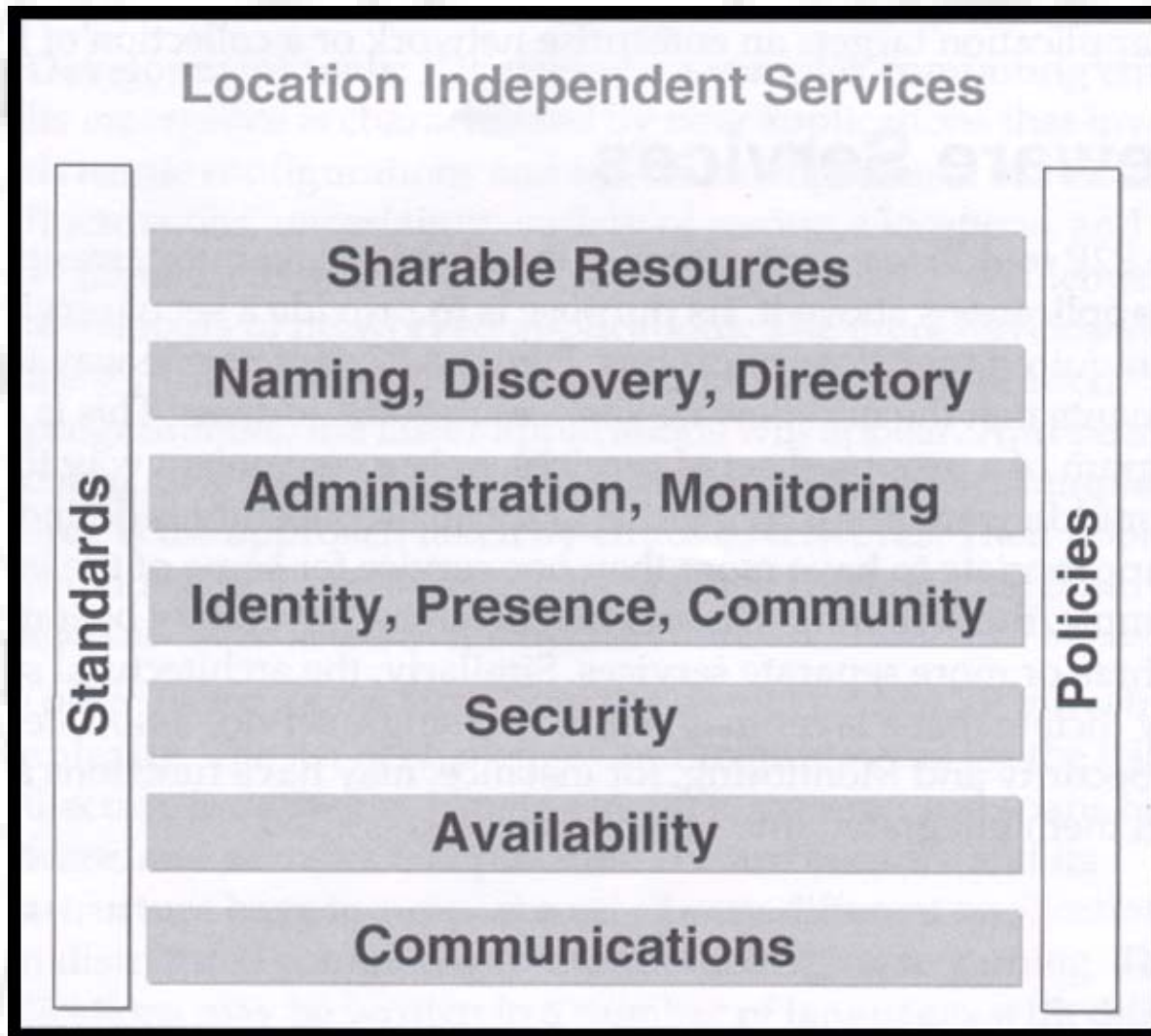
The 3 Vectors of P2P Computing



P2P Infrastructures

- Also referred to as Middleware in a P2P context
- Middleware serves to "glue together" or mediate between two separate programs
- P2P systems must accommodate programs with different underlying infrastructures
- Main considerations:
 - Cross-platform
 - Interoperability

Components of P2P Middleware



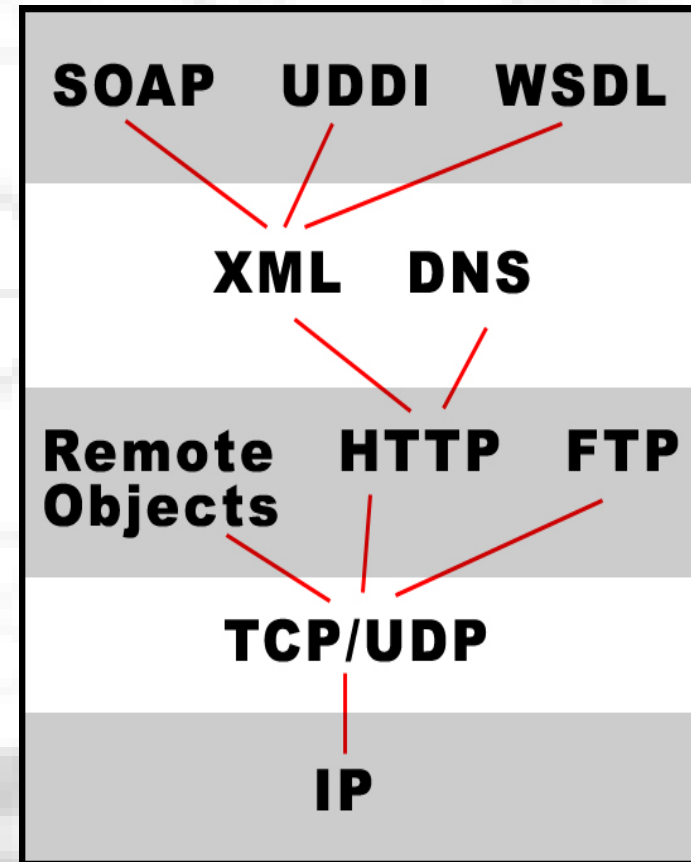
P2P and the OSI Model

OSI Model	
7	Application Layer
6	Presentation Layer
5	Session Layer
4	Transport Layer
3	Network Layer
2	Data Link Layer
1	Physical Layer

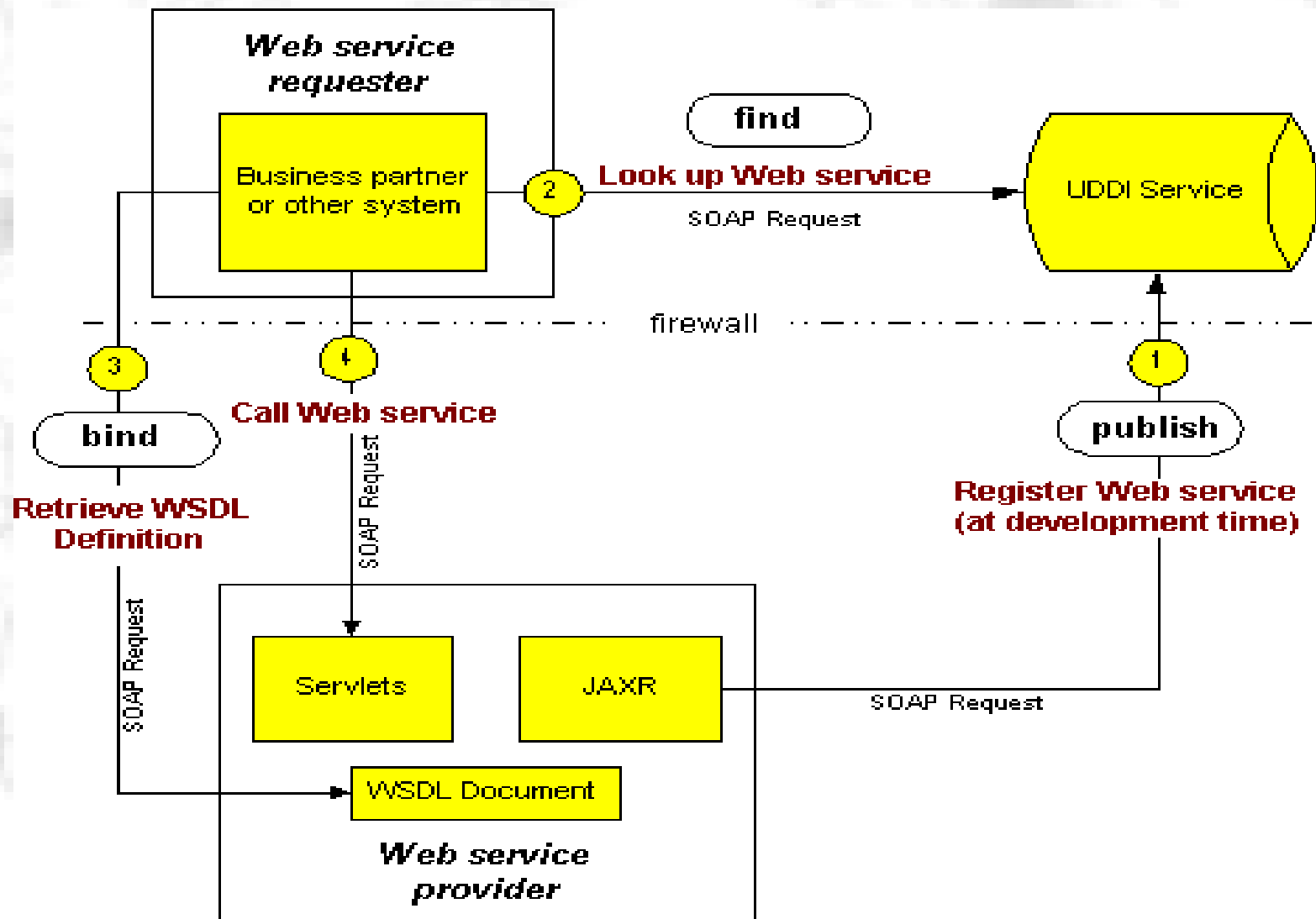
TCP/IP
HTTP DNS
FTP SMTP Telnet
TCP UDP
IP

Enabling Technologies

- HTTP – Hypertext Transfer Protocol
- DNS – Domain Name System
- XML – eXtensible Markup Language
- SOAP – Simple Object Access Protocol
- UDDI – Universal Description, Discovery and Integration
- WSDL – Web Services Description Language
- Remote Objects & Methods – RPC, RMI, CORBA, DCOM



How do they fit together?



Specialized P2P Platforms

- No single P2P platform that developers and users rally behind
- Instead, platforms that target specific areas of applications have evolved

Target Area	Primary Goal
Distributed Computing	Complex problem solving through peer cooperation
Content Sharing	Maximize community size and content validity
Collaboration	Minimize response time in peer communication

Distributed Computing Platforms

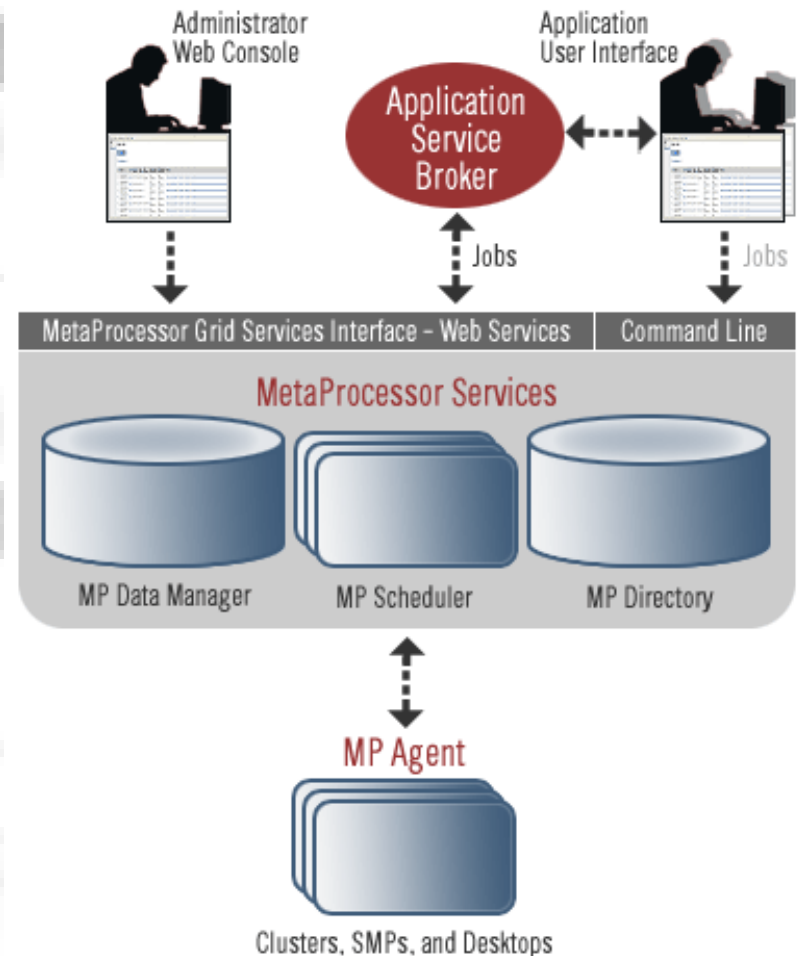
- **The Problem:** Consider an application with a single dataset executed once which returns a single result
- **Goal:** Parallelize the application.
 - Divide and Conquer approach
 - Entire or partial application resides on peers who are given some chunk of data to work with
 - Involves trade-off's between task size and network bandwidth
 - Examples:
SETI@home, Google Toolbar, Simulators



Distributed Computing Example: United Devices MetaProcessor

Basic philosophy: Very few machines fully utilize their network, storage and computing resources

Solution:
The MetaProcessor, allows a quiet, non-intrusive tapping into unused power and resources



MetaProcessor Technologies

- Communication: uses HTTP over TCP/IP
- Firewall/Network Traversal: Intermediate Agent
- Security: uses authentication, Triple DES encryption, and digital signing
- API: Proprietary API's written in C/C++
- Conclusion: Secure and efficient, but it's very "closed" nature and application-specific design choices make it difficult for interoperability

Content Sharing Platforms

- Key activities:
 - Messaging
 - File sharing
 - Caching and Edge services
 - Content distribution
 - Knowledge management
 - Content management
- But what requirements does this impose on a Content Sharing infrastructure?

Content Sharing Platforms

- Basic Requirements:
 - Many same basic requirements of distributed computing, except for managing CPU cycles
 - Unified naming for all peers & files
 - Peer Discovery
 - Complex Content Discovery (searching) mechanisms
 - Support multiple peer groups



Content Sharing Example: The Gnutella Network

- Gnutella is an open protocol for distributed searching
- Peer communicate directly through sockets
- Communication consists of a set of descriptors used for communicating data between servants and a set of rules governing the inter-servant exchange of descriptors
- File transfers use the HTTP protocol

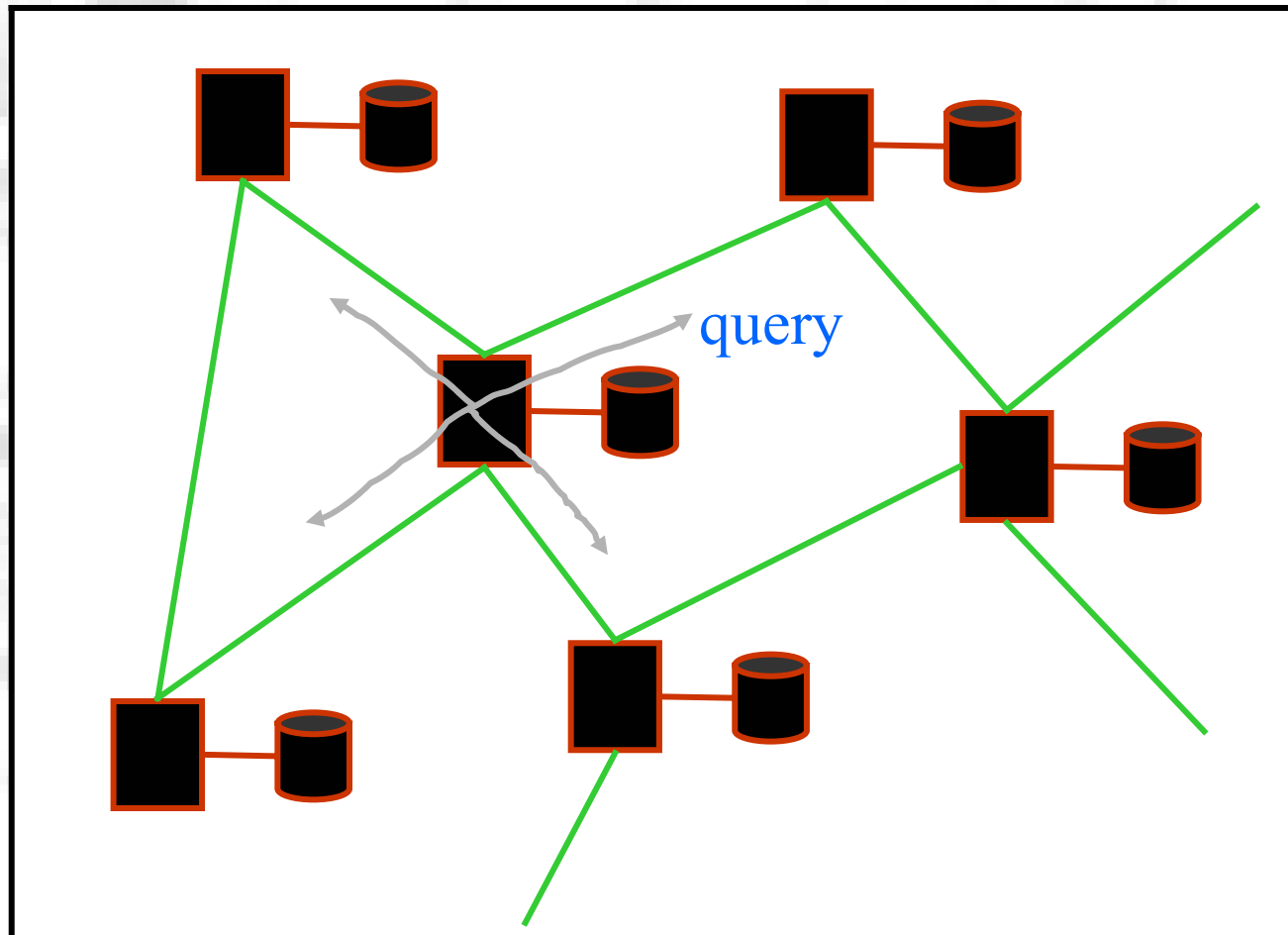
The Gnutella Descriptors

Type	Description	Contains
Ping	Request for a host to announce itself	No body
Pong	Reply to a Ping	IP and port of responding host; Number and size of files shared
Query	Search request	Minimum speed requirement of the responding host; Search string
Query Hits	Reply to a query	IP and port; Speed of responding host; Number of matching files and their indexed result set

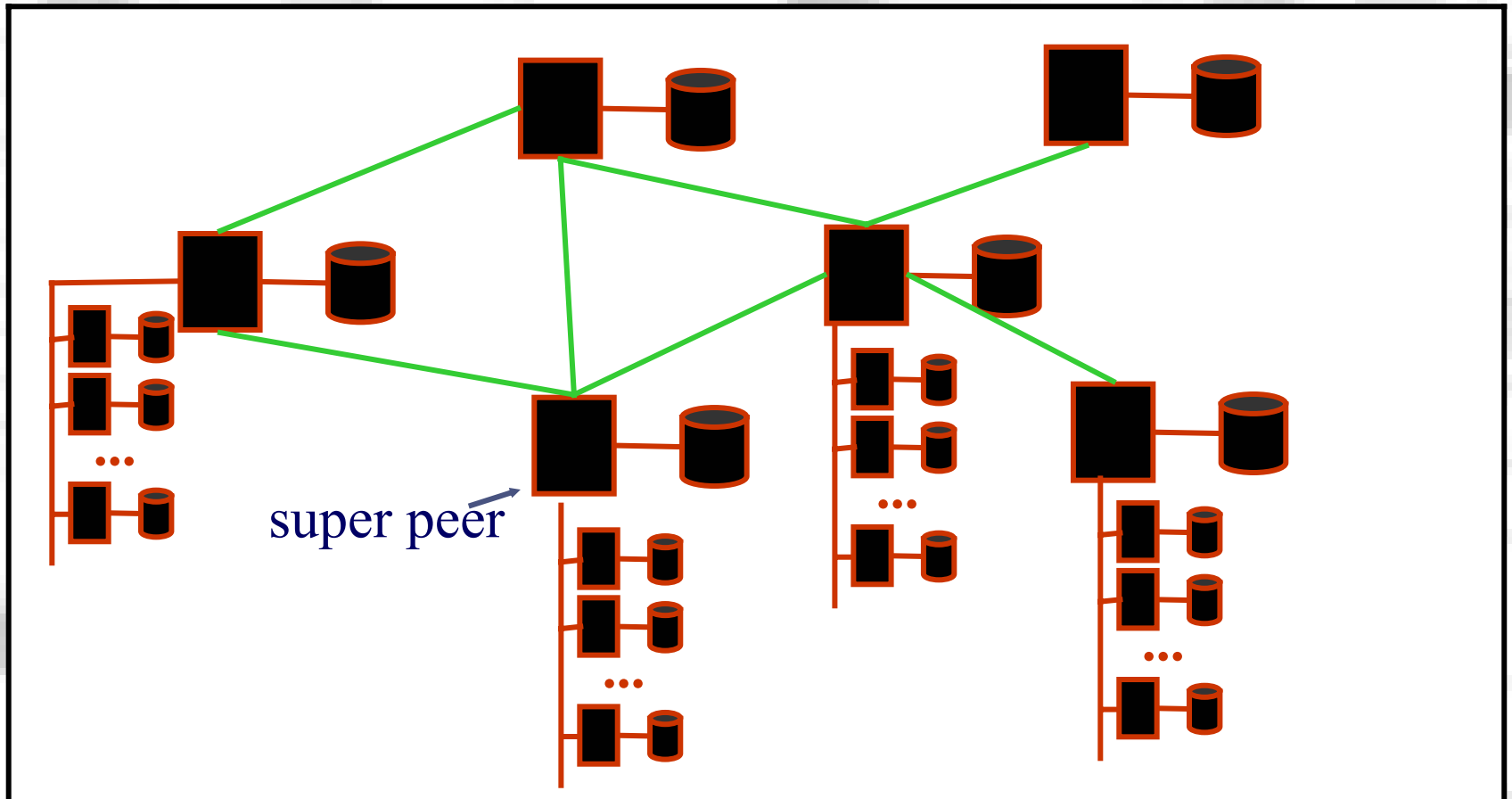
Sample Gnutella Session

Peer 1	Peer 2
GNUTELLA CONNECT/<version>\n\n	GNUTELLA OK\n\n
{send identity}	{receive identity} {send info about other peers}
{receive info about other peers} {start receiving messages from other peers}	
GET /get/<File Index>/<File>/ HTTP/1.0\r\n Connection: Keep-Alive\r\n Range: bytes=0-\r\n User-Agent: Gnutella\r\n\r\n3 \r\n	HTTP 200 OK\r\n\r\nServer: Gnutella\r\n\r\nContent-type: application/binary\r\n\r\nContent-length: 4356789\r\n\r\n\r\n

Gnutella Network Architecture



Kazaa/Morpheus Network Architecture



Collaboration Platforms

Characteristics:

- Accommodate push/pull style of content exchanges
- Rely on negotiated access with provision for trust and security
- Ongoing interactions and exchanges between peers
- Immediacy of interactions is critical
 - security and data integrity become greater challenges
- Models true human nature – cooperation among individuals (peers)



Collaboration Platform Example: Groove Networks

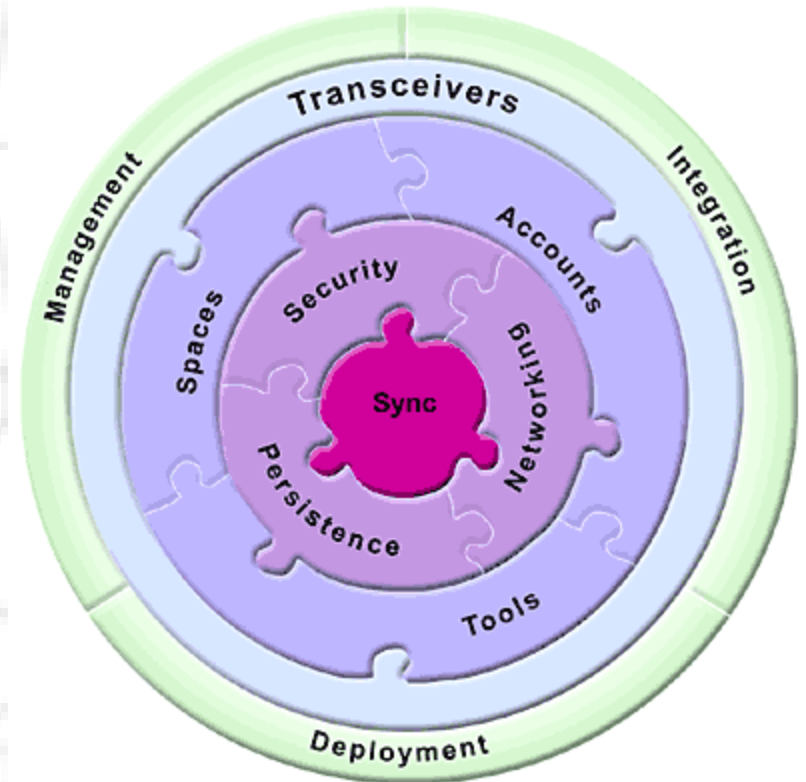
- Groove uses Internet-based P2P model for addressing business demands for online direct and personal communication
- Appropriate for B2B, B2C, C2C
- Groove has 2 main modes:
 1. Immediately useful for general multimedia communication
 2. A platform for developing new P2P applications

Groove Architecture

- Fundamental concept in Groove:

Shared Space:

Each member of the group owns a copy of the shared space.



Groove Technologies

- **Network Flexible:** supports HTTP and SSTP protocols
- **Language Independent:** based on COM specifications
- **Data Persistence:** XML is used for interoperability and extensibility
- **Security:** authentication through public keys, strong encryption, and digital signatures
- **Intermittency:** relay peers are used to temporarily store actions until re-connected

Summary of P2P Platforms

- All P2P platforms address different issues, but common trends emerge:
- Choice of protocols
 - Most rely on IP, TCP or UDP and HTTP
 - Higher levels use XML, SOAP, UDDI, WSDL
- Security
 - Is a function of the needs of the environment that the infrastructure addresses
 - All employ SSL or TLS, but choice of signature and encryption keys vary

Summary of P2P Platforms (cont'd)

- Module Structure
 - Main reason: feasibility of future interoperability
 - Modules can be added/removed
 - Create 1:1 interoperability between different P2P solutions
- Communication Patterns
 - Most employ a hybrid P2P architecture to best utilize all network services and bandwidth

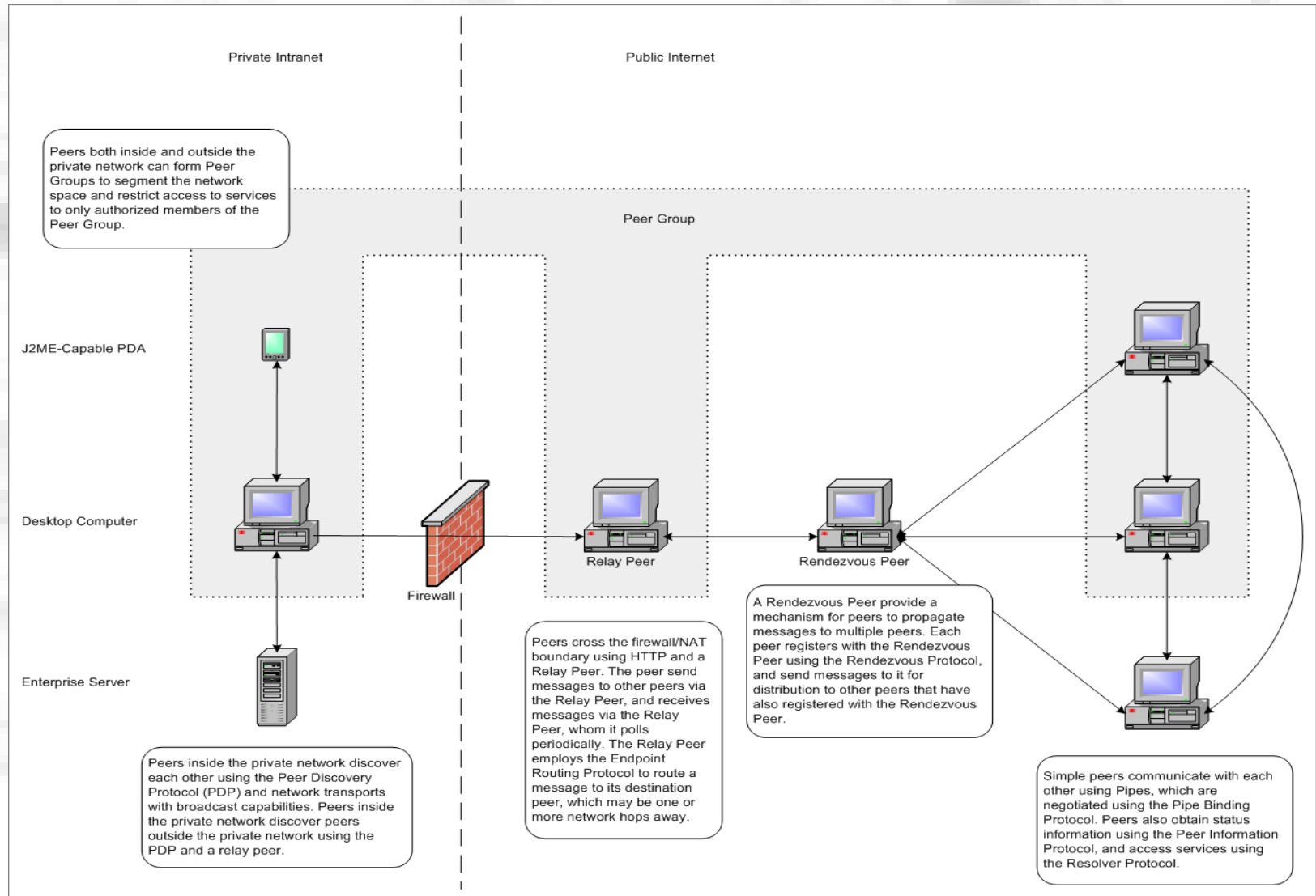
Introducing JXTA

- JXTA technology is a set of open protocols that allow any connected device on the network to communicate and collaborate in a P2P manner.
- Project Objectives
 - Interoperability - across different peer-to-peer systems and communities
 - Platform independence - multiple/diverse languages, systems, and networks
 - Ubiquity - every device with a digital heartbeat

Introducing JXTA (cont'd)

- JXTA standardizes how peers:
 - Discover each other
 - Self-organize into peer groups
 - Advertise and discover network services
 - Communicate with each other
 - Monitor each other

JXTA Conceptual Overview



JXTA Platform Architecture

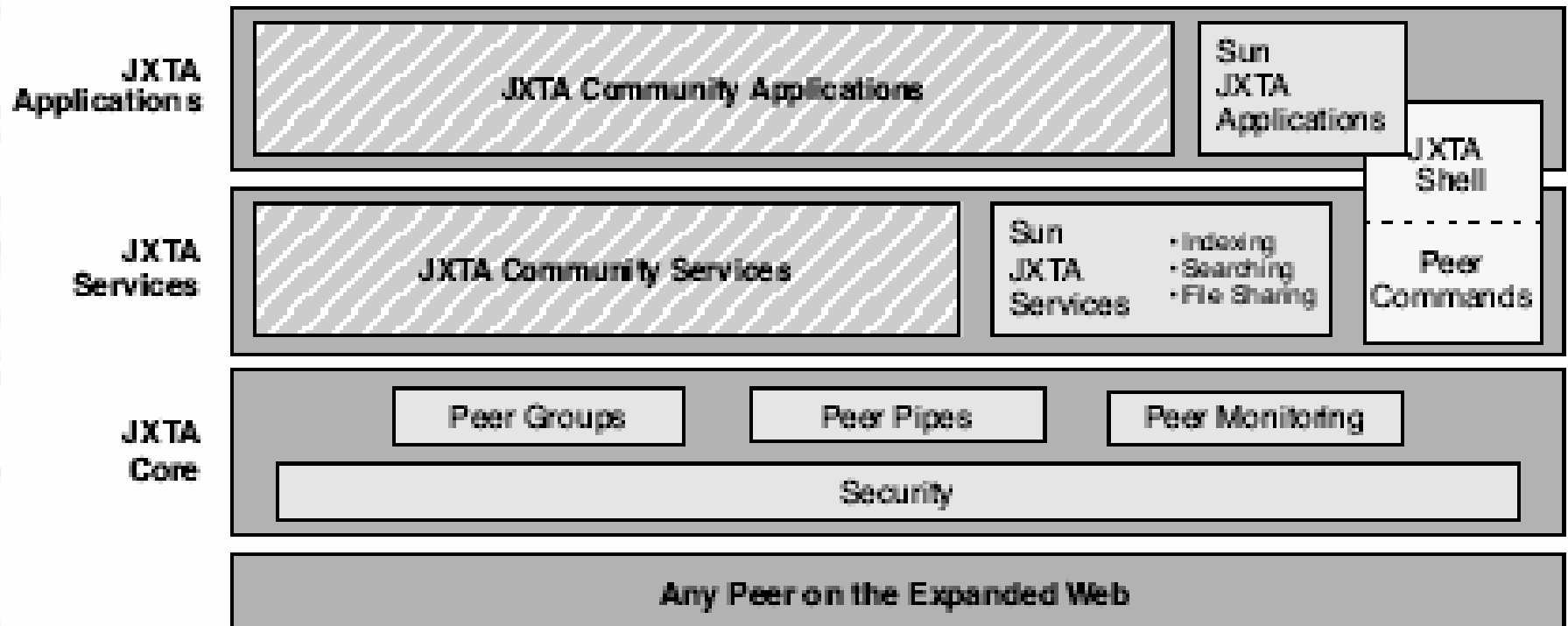


Figure 2-1 Project JXTA software architecture.

JXTA Protocols

- Peer Discovery Protocol
- Peer Resolver Protocol
- Rendezvous Protocol
- Peer Information Protocol
- Pipe Binding Protocol
- Endpoint Routing Protocol

Why JXTA?

- The key benefits of Project JXTA are:
 - designed as a highly modular platform
 - provides simple and essential building blocks for developing a wide range of distributed services and applications
 - specifies a set of protocols rather than an API
 - Innovative mechanisms for addressing critical issues of crossing secure network boundaries

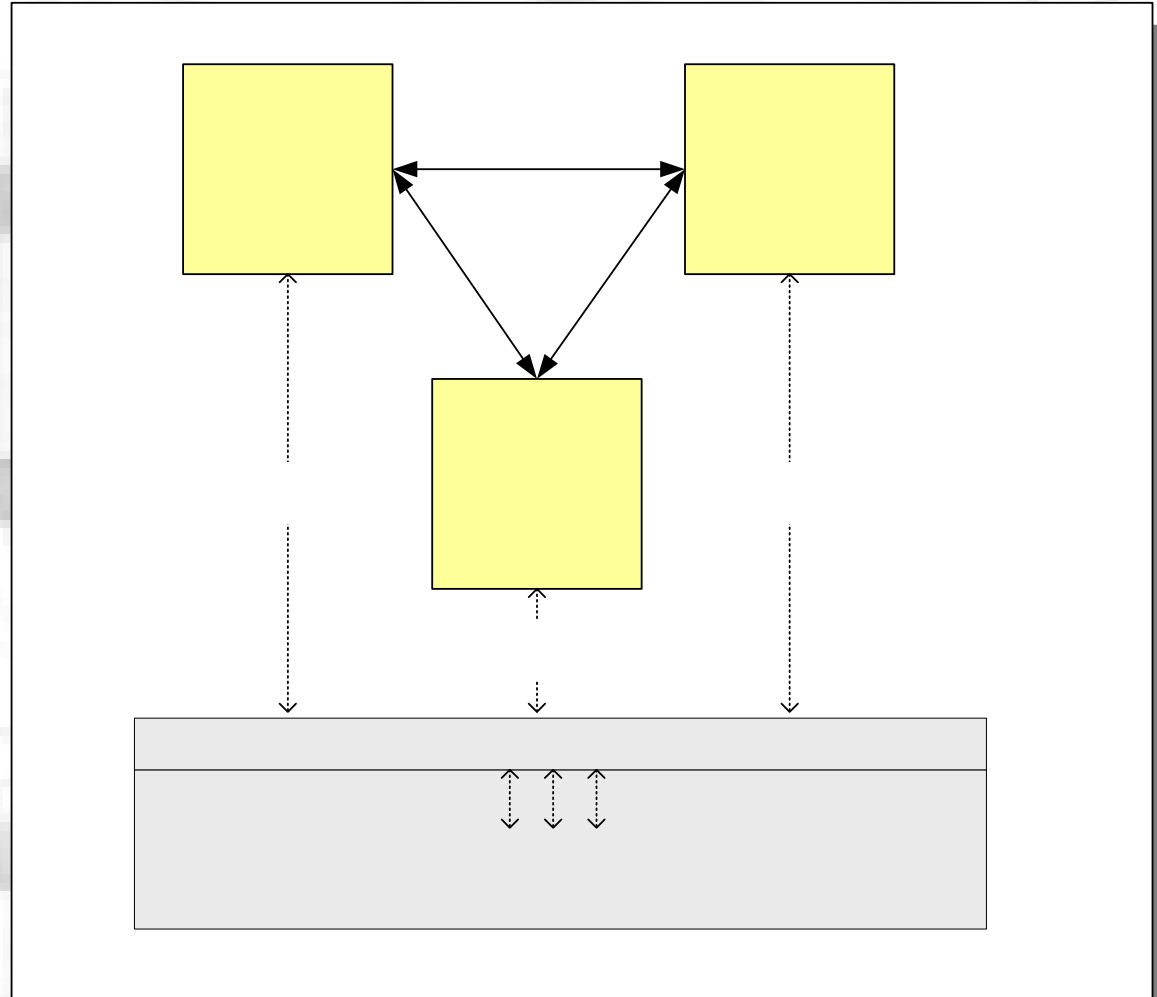
Conclusion

- Technological advancements and global standardization efforts are fueling a P2P era
- P2P best describes the true human nature of interaction – 1:1, peer-to-peer, decentralized
- There is currently no single solution for a P2P computing platform
- Interoperability, not standardization!
- More research needs to be done on the characteristics and properties of P2P systems

QualityChat Demo

What is QualityChat?

- Instant Messaging application
- Employs a Hybrid P2P Architecture
- Created in Java



References

- Peer-to-Peer Computing. Barkai, David. Intel Press, 2001
- JXTA. Brendan, Wilson. New Riders Publishing, 2002
- Modeling Peer-to-Peer Network Topologies Through “Small-World” Models and Power Laws. Jovanovic, Mihajlo. ECECS Department, University of Cincinnati. 2001
- Scalability Issues in Large Peer-to-Peer Networks. Jovanovic, Mihajlo. ECECS Department, University of Cincinnati. 2001.
- Open Problems in Data Sharing Peer-to-Peer Systems. Garcia-Molina, Hector. ICDT Conference. January 2003.
- Distributed Systems Topologies
http://www.openp2p.com/pub/a/p2p/2002/01/08/p2p_topologies_pt2.html
- Project JXTA
<http://www.jxta.org>
- Groove Networks
<http://www.groove.net>
- United Devices
<http://www.ud.com>
- searchNetwork Definition
http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci212769,00.html