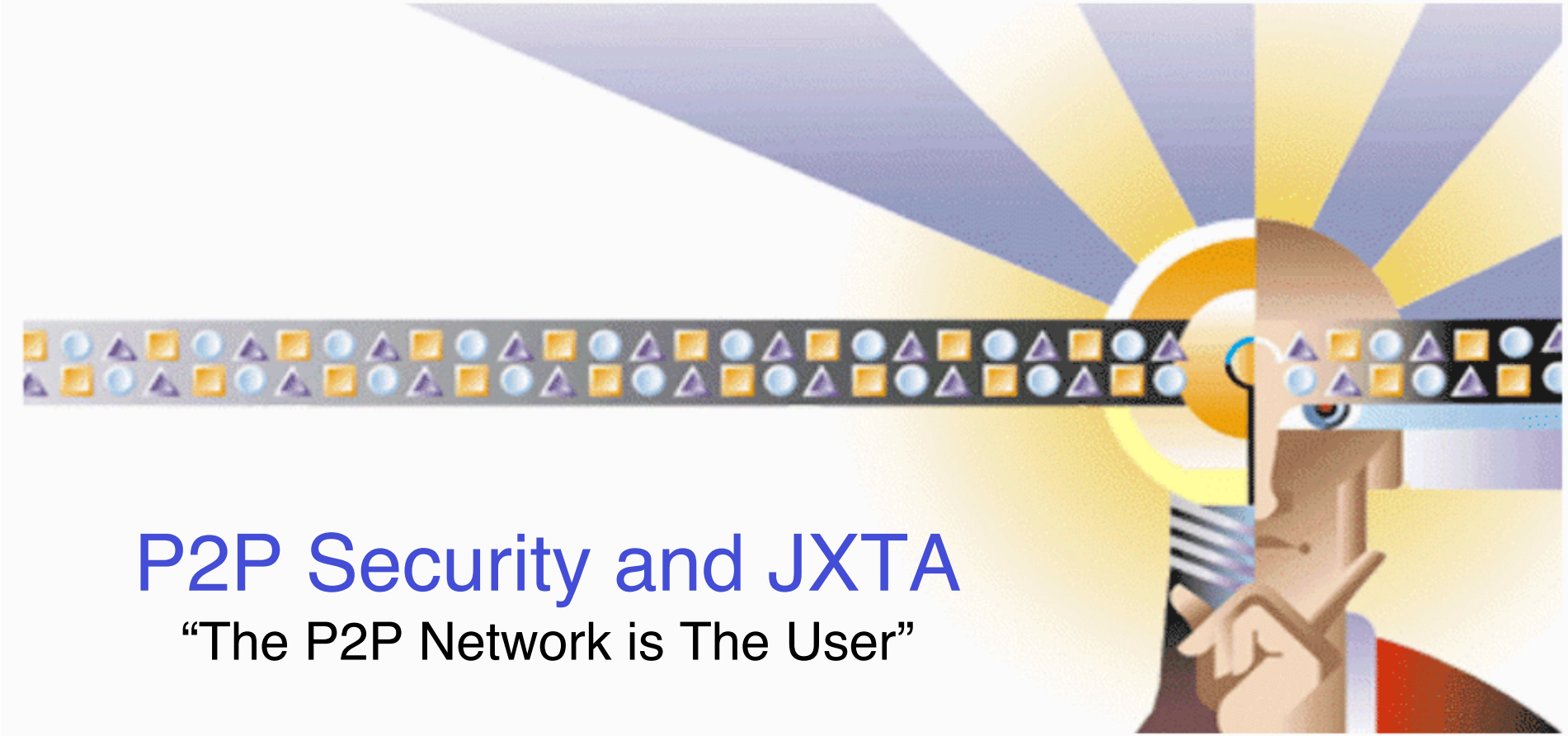


P2P Security and JXTA

"The P2P Network is The User"

Bill Yeager

Senior Research Staff

Sun Labs

Sun microsystems

We make the net work.

Highlights

- Part I
  - A Personal P2P Vision
    - Why P2P Now?
    - The P2P Network *is almost* The User
- Part II
  - Perspectives on Privacy and P2P Security
  - A Brief Introduction to Jxta
  - A Sun Labs Project on the Jxta Infrastructure
    - First: Enterprise Strength Security on Jxta
    - Second: The P2P User's Network Presence
      - The P2P Network *is* The User

A Personal P2P Vision

The P2P Network is the User

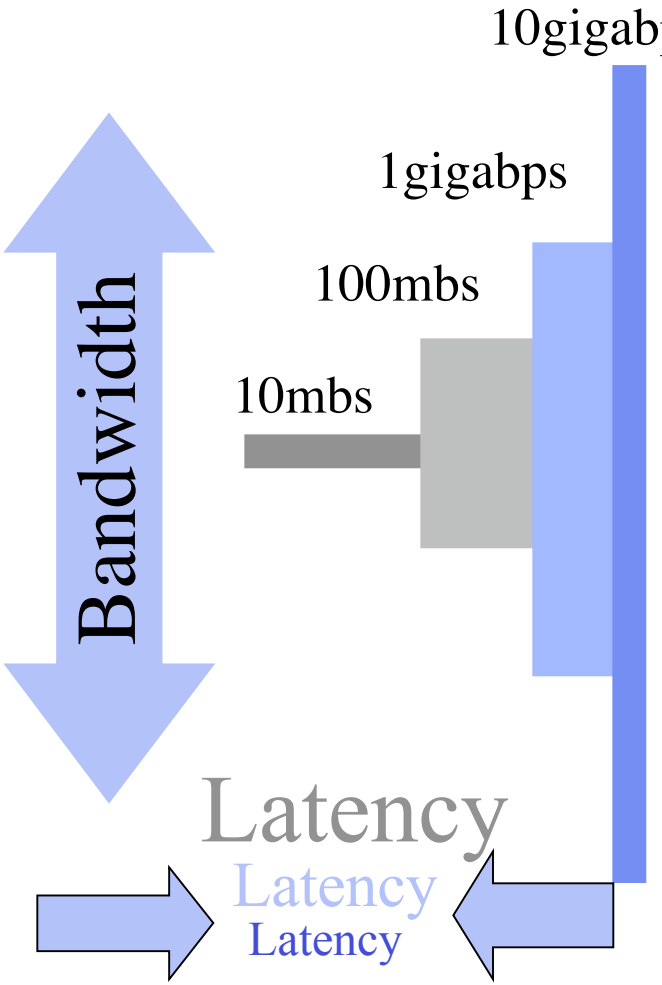
Why P2P Now?

The Evolution of Bandwidth: 1981-1994

- The growth of local transport usually precedes the need for rapid intra- and interstate transport
  - City streets then state and interstate highways
  - Local Area Networks spawned Wide-Area-Networks
    - Businesses followed the University LAN model
    - Then T1 WANs connected LANs to one another sometimes using the Internet
  - These LANs and WANs accelerated the internet's growth
- LAN service was always superior
  - The end **user** ruled, most traffic was LAN based, each user's system had a unique IP address and network communication was end-to-end.

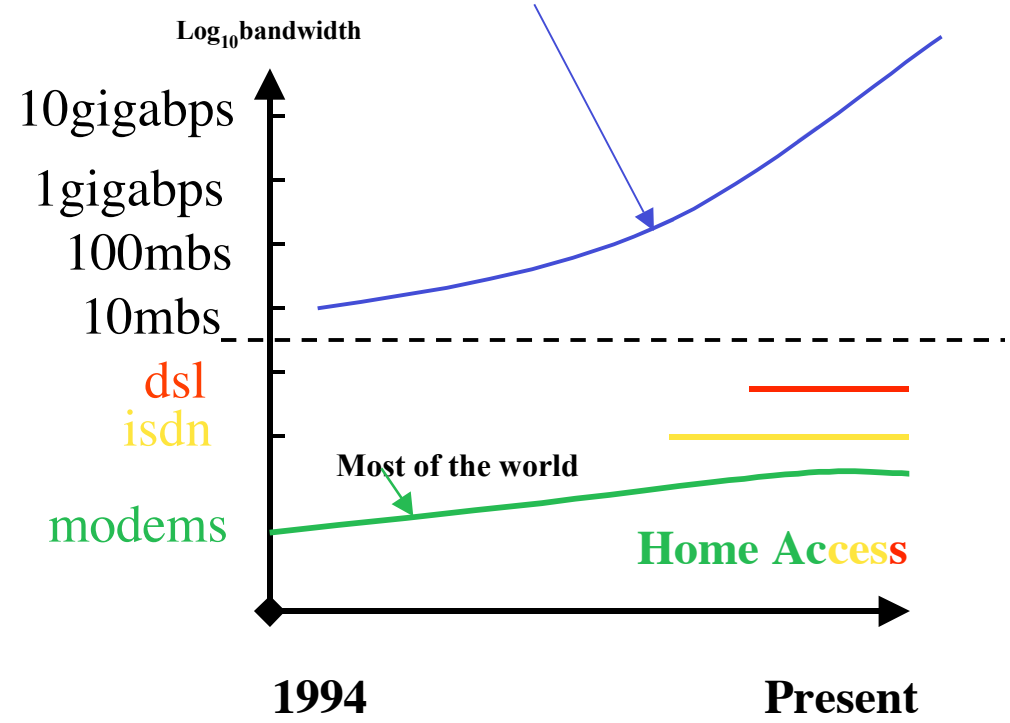
The Evolution of Bandwidth:

1994 to Present - Home access is not getting a fair share



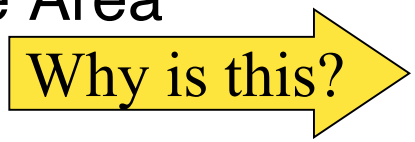
Internet:

Less latency, higher bandwidth

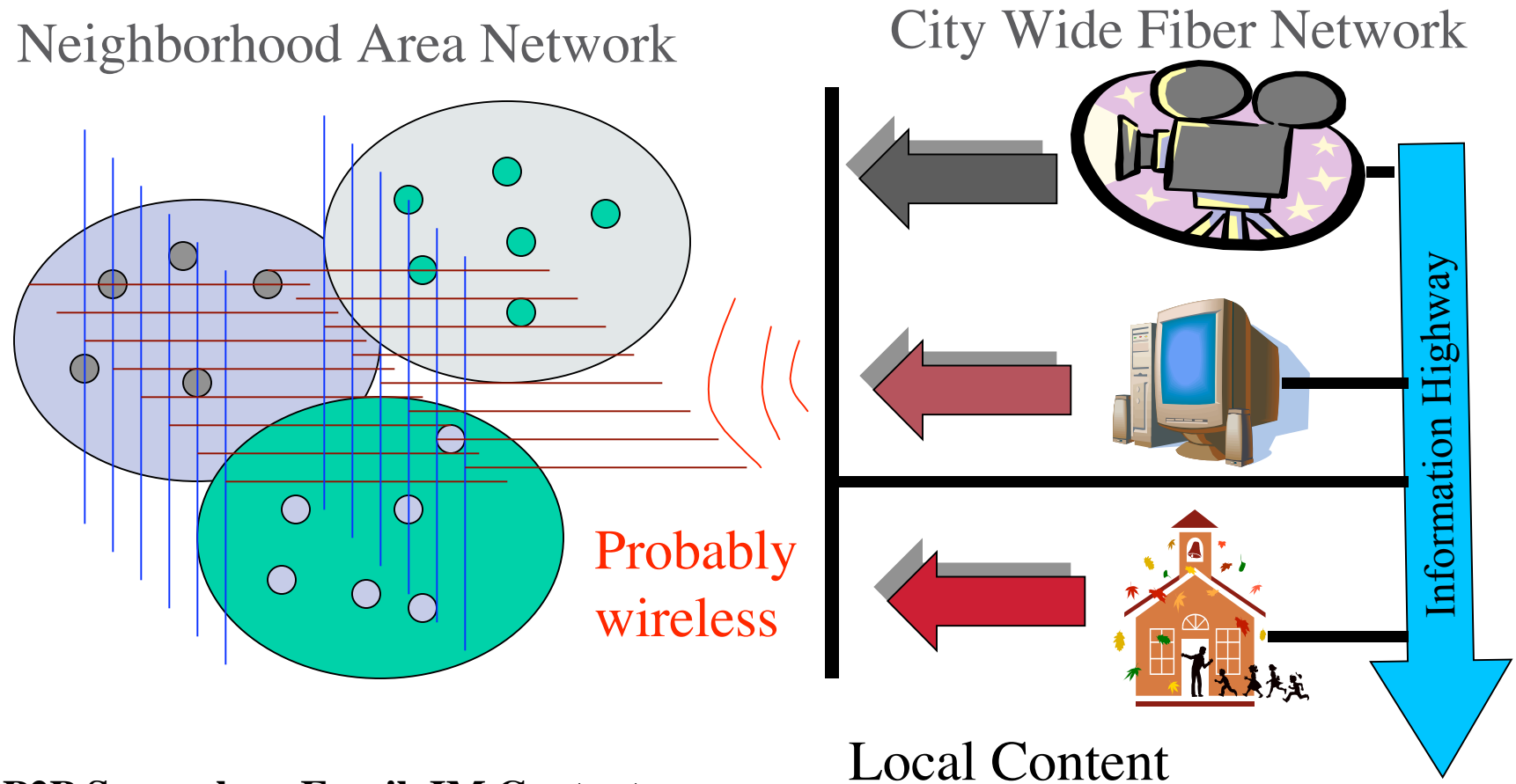


On the P2P Internet

- On the Internet if I want great pinot noir I need to go to Bourgogne via http/GET on The Information Highway
  - No local wine merchants in my Internet neighborhood
- The P2P Internet will
  - Migrate the content close to the users of that content
  - Increase neighborhood bandwidth/access
    - Wi-Fi (802.11a/b/g) Hot Spots
  - Connect neighborhoods to City Wide Area Networks, etc …



P2P- Put More Bandwidth Where the Action Is: In The ***Users***' Neighborhoods

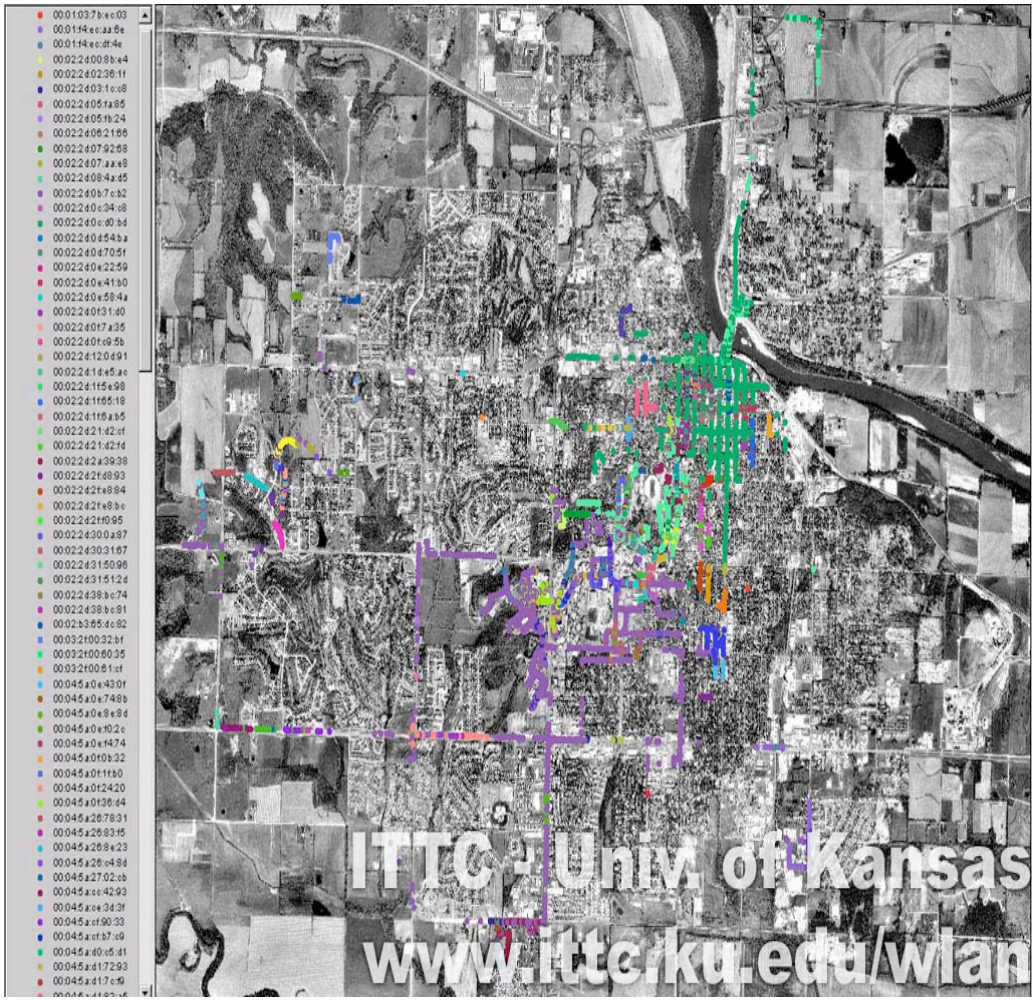


P2P Server less, Email, IM,Content Sharing, Collaboration, etc …

The P2P Internet Will Reflect Users' Behavior and Requirements

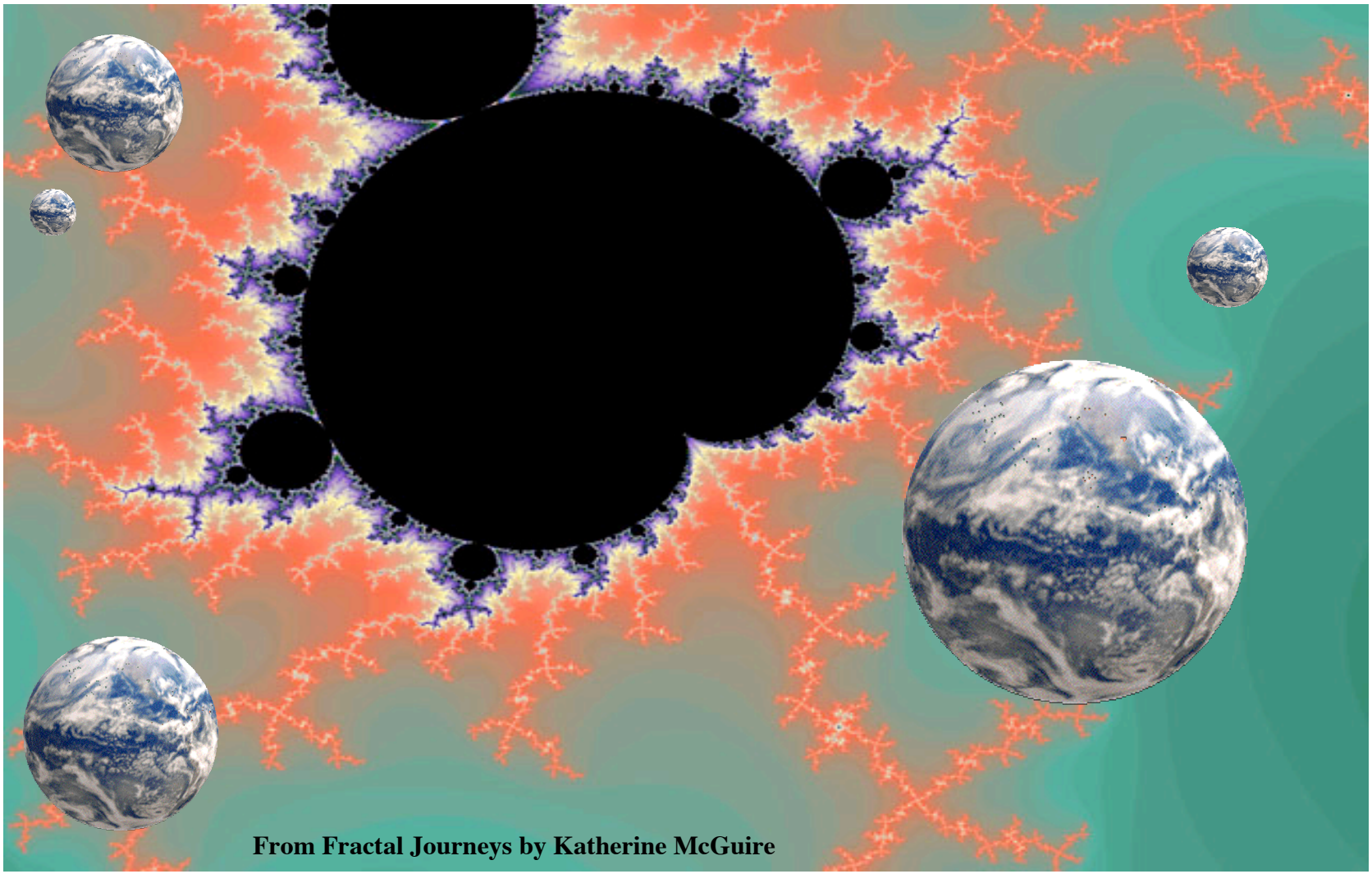
- Sometimes ad-hoc and sometimes organized
  - Spontaneous connected communities will arise
    - These are social networks and will reflect the best and worst of human interactions
  - Anonymity, privacy and fair use of content will be respected
- Single points of failure will go unnoticed since availability is built-in
  - Servers will feed the content supply chain
  - Content will be accessed as close to home as possible
    - If one wishes to see their neighbors photos of a recent trip to Sweden, and one has access rights, no-intermediary storage will be required
      - A simple "drag and drop" between cooperative neighbors' P2P systems will do the job

It's Already beginning: 802.11/b Hot Spots at Lawrence, Kansas



The P2P Internet: Super Hot on the Edges

Where the User Lives



Perspectives on Privacy and P2P Security

Security by Design - A Good Point of Departure

Those who design systems which handle personal information therefore have a special duty: They must not design systems which unnecessarily require, induce, persuade, or coerce individuals into giving up personal privacy in order to avail themselves of the benefits of the system being designed.

Leonard Foner, MIT Media Labs

Design Secure Software for Ordinary Human Beings

- People understand locks, burglar alarms, eaves dropping, peeping Toms, identity theft, ID Cards, secrets, etc…
- People do not understand the “man-in-the-middle attack,” public key cryptography, X509.v3 certificates, etc …
- People must know their systems can be compromised and that

Next Slide

Secure The Infrastructure in Human Terms

- These threats must be stopped in their tracks
  - Eaves dropping or wire-tapping
  - Identity theft
  - Forgery
  - Unreliable transactions: You ordered a book and received tennis balls
  - A person claiming they did not borrow money from you, or did not bet on that football game
- We didn't mention the network. Just the familiar problems …

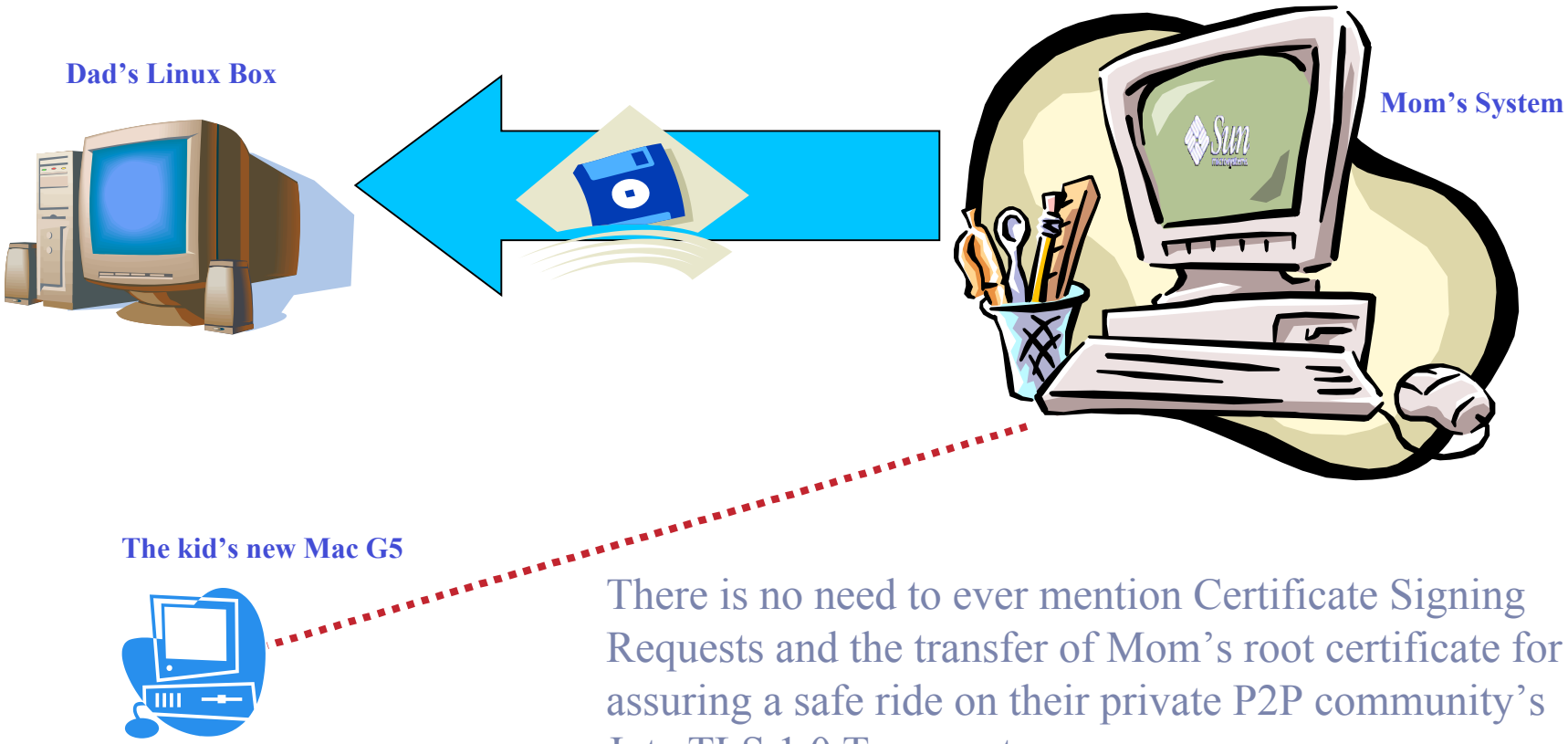
Security is a Social Phenomena we live with every day of our lives

Let's design our software to reflect this perspective pointing out that a trip on the information highway also requires safety belts, airbags, and car seats for children

The P2P Security Toolbox

- We need a boss, a responsible party, someone whose signature we must trust (Our Certificate Authority)
  - Dad, mom, your boss, the god father (trust him or die!)
- Or, we need a group of friends whose signatures we trust given their reputation (PGP)
- Or, certified, well known, certificate authorities as guarantors
- Users don't need to know about public/private keys
  - They only need to know there are guarantees by someone they trust that the bad guys will be stopped

Here the Trusted Party Assures our Security Out-of-Band



There is no need to ever mention Certificate Signing Requests and the transfer of Mom's root certificate for assuring a safe ride on their private P2P community's Jxta TLS 1.0 Transport.

In Summary

Our security toolbox must meet the real requirements of the social contexts of P2P communities. It must contain a full collection of sockets, and a socket wrench that permits the user to apply the right amount of torque to yield a comfortable sense of security.

It must come with an instruction book that explains the risks in terms that are clear to the user and honestly reflect the capabilities of the software we create.

Finally, I would hope that in some not too distant future there will be independent quality of assurance ratings of software in the terms we have been discussing.

A Brief Introduction to Jxta

The Jxta Platform

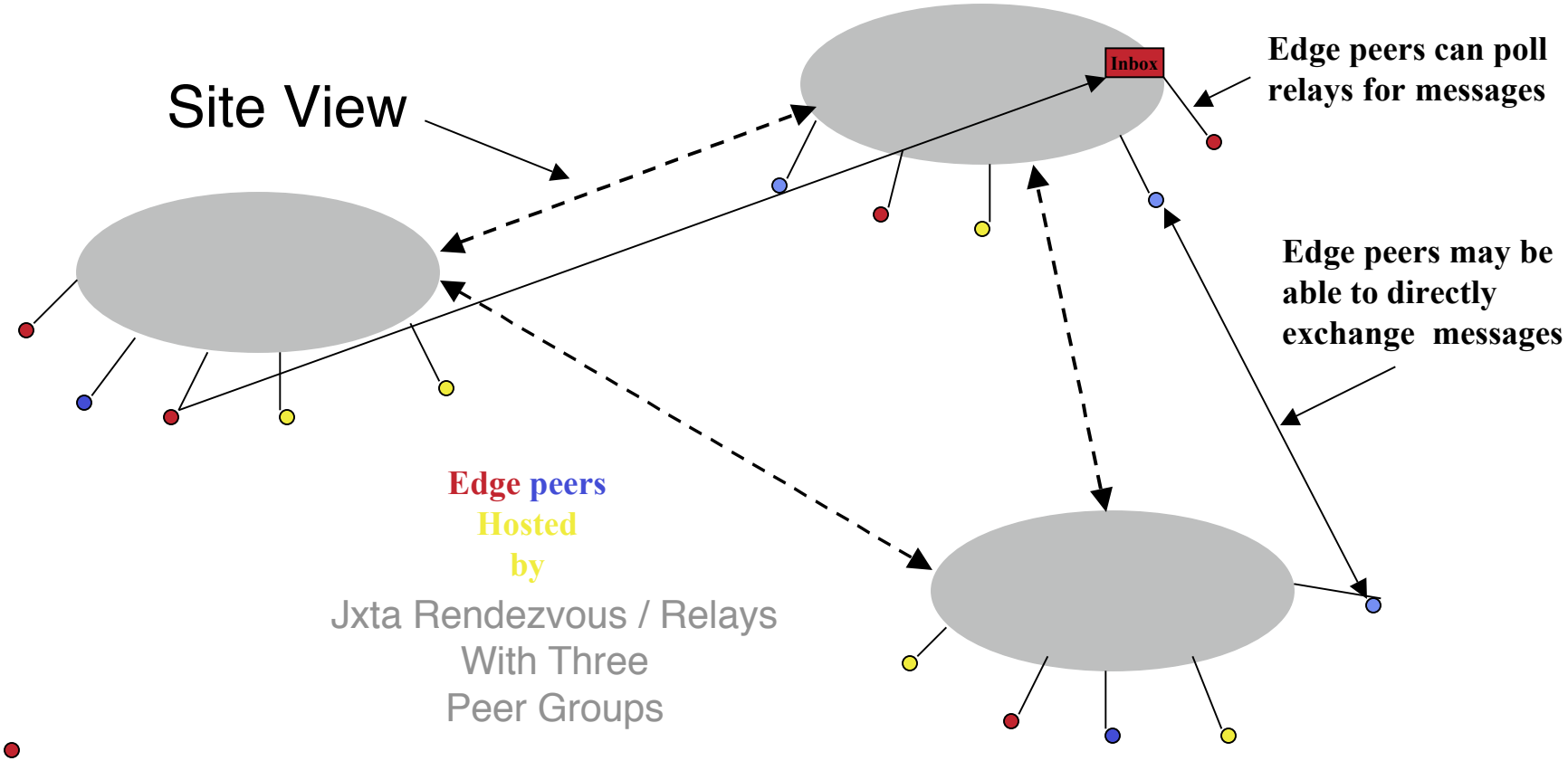
- The Jxta Platform is a set of protocols and services whose implementation permits peers to
  - Discover and communicate with one another in a way that is independent of both location and the underlying network infrastructure
  - Be clients and servers
    - Peers can initiate connections to one another
  - Send confidential, authenticated data (TLS 1.0)
  - From private, connected communities called peer groups

Jxta Platform Components

- Edge Peers
  - Desktops, Laptops, PDA's, Mobile Phones, Bluetooth enabled devices, Server clusters, etc …
- Rendezvous
  - Host edge peers
  - Support a distributed lookup service (Currently the Chord distributed hash table is used)
  - Together form the rendezvous site-view
- Relays
  - Route messages between peers that do not have direct real network connectivity
    - For example: Edge peers may be NAT or firewall limited

The Jxta rendezvous and relays

Rendezvous are almost always relays



- Members of the red, blue and yellow peer groups

Peer Identifiers

- Peers use the underlying platform protocols to self-organize into an “overlay network” using unique identifiers called peerID’s
- At platform boot time peers dynamically create/update XML peer advertisements containing
  - The unique peerID (fixed)
  - The peer’s name (fixed and not necessarily unique)
  - With the current real transport information
- These advertisements are published on the peers’ rendezvous

Pipes: Overlay Network Communication

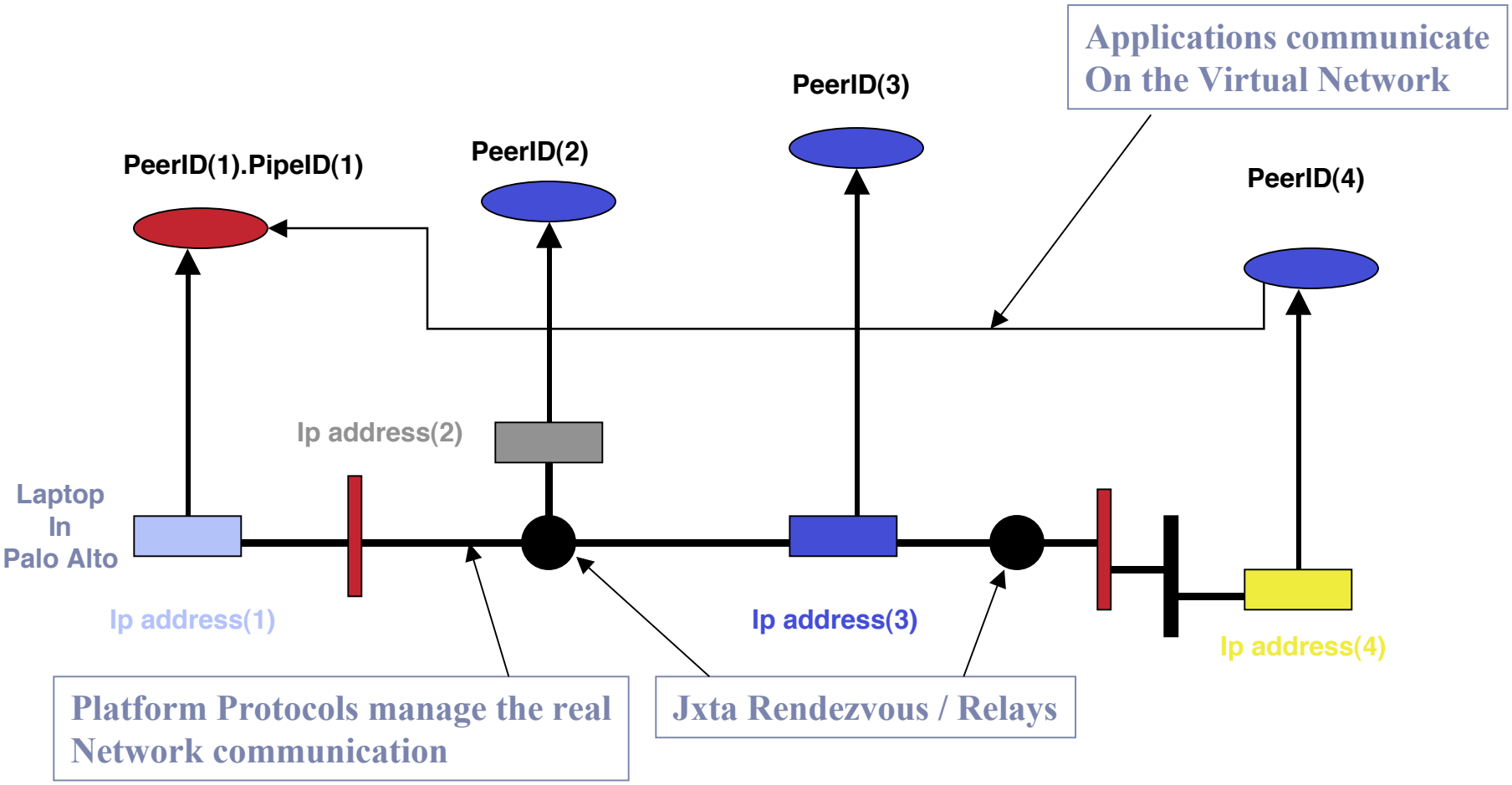
- Applications use pipes to communicate on the overlay network
  - They are described in XML advertisements
    - Pipe name - Fixed and not necessarily unique
    - Unique pipe identifier - Fixed
    - Pipe type
      - Unicast [Secure], Bi-directional [Secure]
      - Propagate or Multicast
  - The pipe advertisements are published on the rendezvous site-view, for example:
    - Hash("PIPENAME" + Pipe's name) yields the rendezvous where the {"PIPENAME" + Pipe's name, peerID(content source) is found
- Pipes can be viewed as overlay network virtual ports
- PeerID.PipeID is a unique communication endpoint or virtual socket

Peer/Pipe Discovery

- On the overlay network peers discover on another by *looking up* the peer that has a copy of a peer advertisement associated with a unique peer, for example:
  - Hash("PeerName" + peer's name) yields the rendezvous where
    - {"PeerName" + peer's name, peerID (content source)} can be found
    - The peerID(content source) yields a route to this peer, and the query for the advertisement is sent to that peer
- While the real transport information may change over time, the peerID and peer name are fixed
- Pipes can be discovered in the same manner

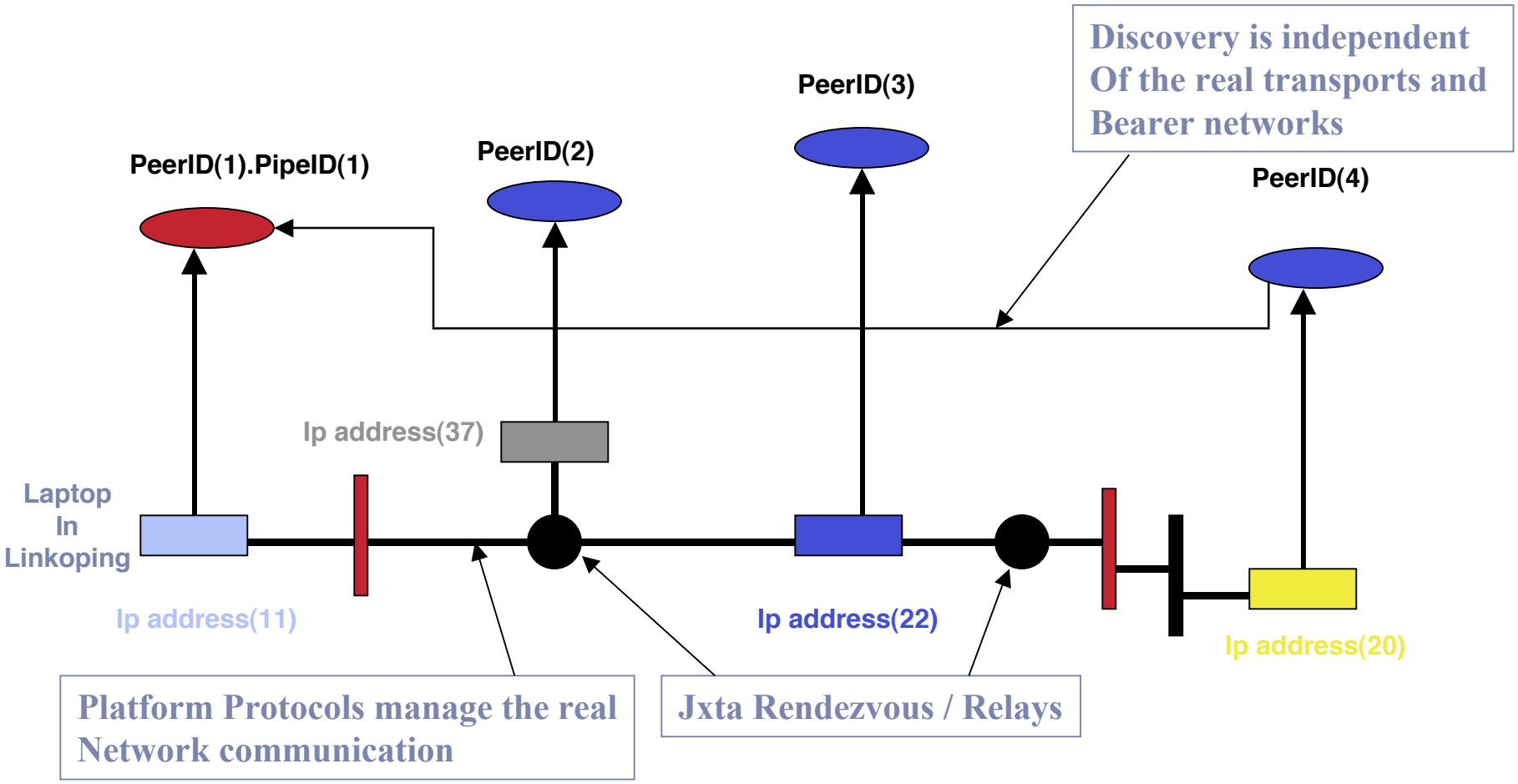
Overlay Network: Jxta Virtual Network

- Peers use the underlying platform protocols to self-organize into an “overlay network” using peerID’s



Cool feature: Implicit Mobility!

Peers can connect anywhere on a Jxta Virtual Network as long as they can find a rendezvous.



Enterprise Strength Security The Sun Labs Jxta Infrastructure

The goal is security sufficiently strong to do business in the enterprise on the one hand, and, on the other hand, to create a model for Jxta security that can be morphed to work on non-enterprise Jxta infrastructures.

The peer nodes will run pure Jxta protocols and any jxta platform changes will be returned to the open-source community. We plan to also place our new security code there.

Leonard Foner's security principles are our stake in the ground. We have applied them at every layer in our design and implementation from the infrastructure to the applications.

We do not invent new security protocols, rather we apply the known protocols to fix the insecure aspects of the Jxta protocols, and to guarantee that a User's data is absolutely private …

I am going to discuss the two security holes in Jxta. These have been known to be there since the beginning and the software has been designed in such a way that one can “override” these difficulties. Open-source work is in progress in this area. And the implementation is in JAVA after all, and one thus begins with sandbox security.

The Jxta security model is general to satisfy ad-hoc, P2P network requirements, and one only needs to use a larger socket from our security toolbox to tighten things up to satisfy the most stringent constraints.

One can view these weaknesses as a strength since the Jxta software is extremely flexible and adaptable. It was designed to give $0 cost security to the average person. The default crypto suites are extremely strong.

This will be clear as we proceed from the problems to the solutions.

What are the Weak Points We Must Fix?

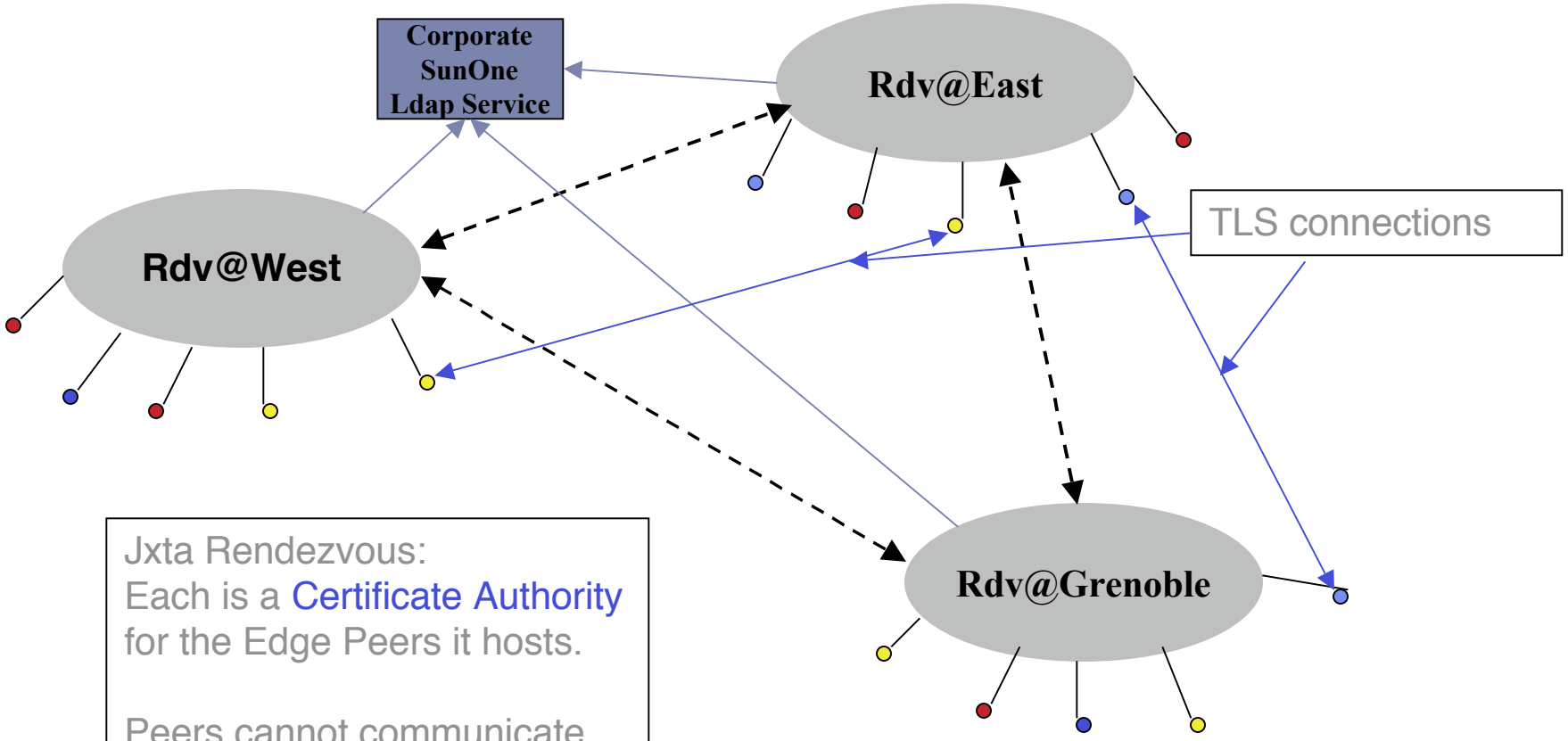
1:

- Neither Peer Names nor Pipe Names are guaranteed to be unique and peer and pipe advertisement lookup uses names
  - Thus, any peer can acquire a copy of another peer's advertisements and, thus, become an imposter, steal that peer's identity
    - Jxta does use Cryptographically Based ID's that when used with an out-of-band fingerprint verification can prevent this theft.
    - This weakness can, if anything, raise havoc and when duplicate advertisements with different peer node sources appear. Communication becomes unpredictable.
    - Advertisements with duplicate names and different ID's can also make communication unpredictable because the lookup is then ambiguous.
- Simple solution:
  - Do not permit duplicate names in advertisements
    - Applications can set these names
  - Make each peer authenticate itself *in-band* as the owner of its unique pipe advertisement before communication is possible.

What are the Weak Points We Must Fix? 2:

- In the default implementation root certificates are distributed *in-band* in the peer advertisements since each peer is its own certificate authority
  - This permits a “man-in-the-middle” attack
    - This is difficult to accomplish given the lookup mechanisms
      - Imagine someone takes over a router, or creates a renegade rendezvous/relay. The code is open-source after all.
  - Note that this is in fact *OK security* in the right social context
    - Simple chat applications, inter-office peer groups, familial conversations, anonymous, independent Jxta connected communities
    - The default crypto suites are strong: RSA1024, 3DES or RC4, SHA-1, MD5
- Simple solution for the enterprise:
  - *Do not* distribute root certificates in-band or out-of-band
  - *Do not* permit peers to be their own certificate authorities
  - Make each peer authenticates itself *in-band* as the owner of its guaranteed unique pipe advertisement before communication is possible.

The Sun Labs Jxta Infrastructure



Jxta Rendezvous:
Each is a Certificate Authority
for the Edge Peers it hosts.

Peers cannot communicate
with one another without
having first been issued a
Rdv-CA signed cert.

Details

How Did I Implement These Simple Solutions?

1:

- We have three rendezvous and each has been turned into a certificate authority
- Our software arrives with the certificate authorities root certificates in one of the jar files
- The first time the Jxta platform is started with our applications (to be described later)
  - The user logs in with a unique userID and the Sun password
  - Under TLS, this peer connects to its unique rendezvous, generates a public/private key pair and a PKCS#10 Certificate Signing Request which is roughly like:
    - SubjectDN: O=Sun.com, L=MTV29, C=US, CN=Unknown OU=Sun Labs
    - Public Key: Lots of bytes
    - PKCS#1 RSA signature of the above using the private key

No Imposter Attacks

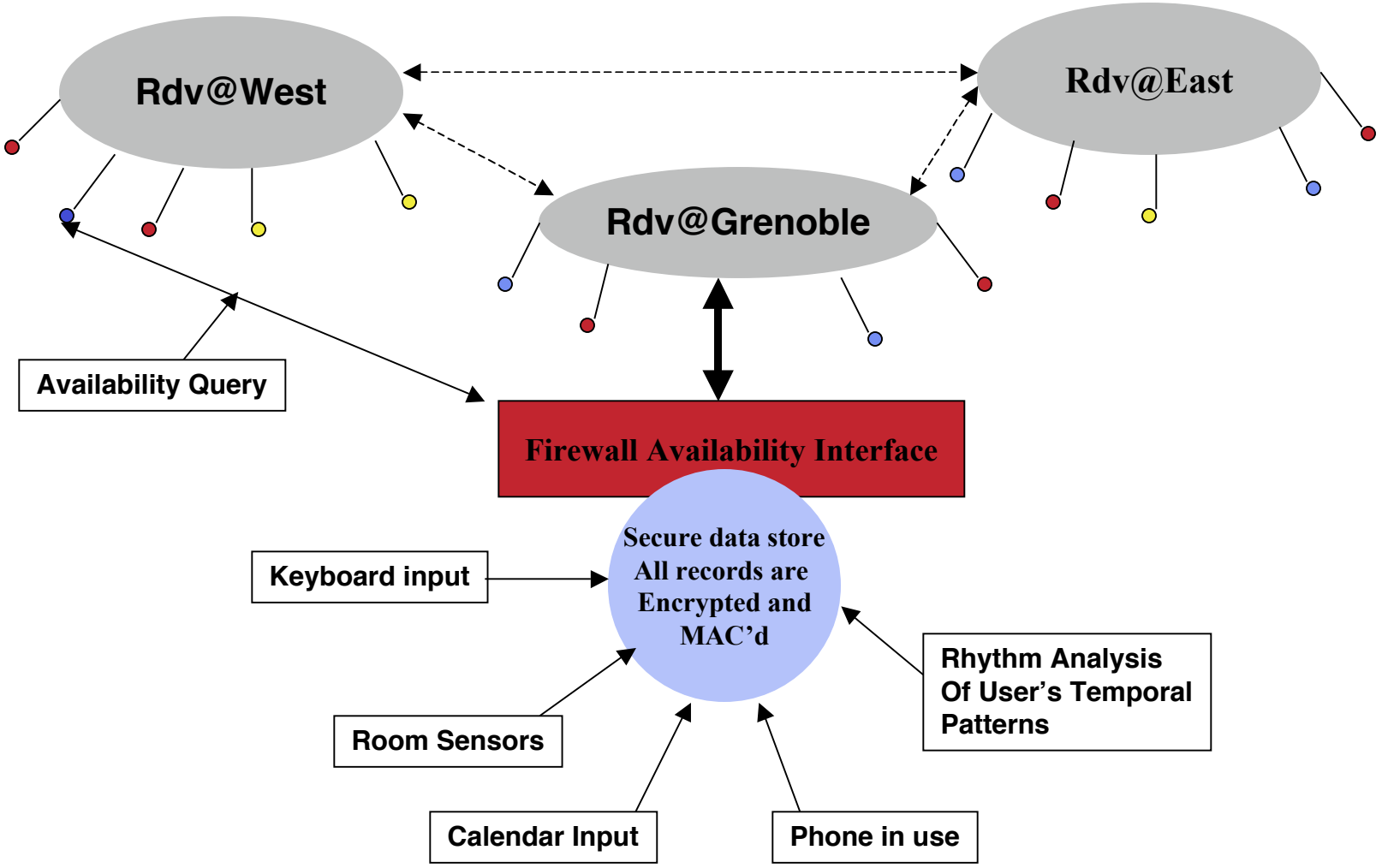
- This along with the userID and password is sent to the rendezvous
- The rendezvous verifies the PKCS#10 CSR private key signature using the included public key in the CSR
- Authenticates the user with the corporate LDAP service
- If the authentication succeeds:
  - Retrieves the user's email address from the LDAP directory
  - Creates an X509.V3 certificate substituting:
    - CN="William.Yeager@Sun.Com" for the CN=Unknown in the SubjectDN of the issued certificate
  - The Common Name in the issued X509.v3 certificate is used as a suffix to every Pipe name in every pipe advertisement created by the application for "William.Yeager@sun.com"
    - Thus: "KaperAgentService.William.Yeager@Sun.com" is a unique agent listening pipe name. This pipe advertisement is for the exclusive use of the possessor of the public/private key pair associated with the X509.V3 cert
  - If anyone tries to be an imposter that is defeated in the following way …

Because …

- Since the Rdv-CA signed X509.V3 certificate has
  - CN="William.Yeager@Sun.Com" in the SubjectDN
  - And
    - A) This cert is used to establish TLS connections (all of our P2P communications use TLS)
    - B) Each peer has the root certification of the Rdv-CA issuer
  - If the TLS handshake succeeds, then private key used by the connecting peer belongs to the public key in the cert and
    - A) We know the Certificate was signed by one of our Rdv-CA's
    - B) We extract the CN and compare to the suffix of the pipe Name used for the connection, for example:
      - "KaperAgentService.William.Yeager@Sun.com"
  - If they match, then we are talking to William.Yeager since he is the sole possessor of the private key of the key pair, and the TLS session is with Bill's system.

So, How Do We Use This Infrastructure in our Research Project?

All Peers Have a Personal P2P Network Presence



Privacy of Presence Data is the Guiding Principle

- All queries have access policies bound to the querying peer node
  - XACML (Extended Access Control Meta-Language) is used to define the policies
- Query responses never reveal the aggregated presence data
  - “Can I contact Rita at 11am” might yield
    - No response at all, “Send email,” “Rita is busy,” or “Call her after 11:15” depending on the policy bound to the querying peer node

In a very real sense this personal network presence embeds a reasonable representation of the User's daily activities into the Jxta P2P Network, and in this sense the network is becoming sentient. A real awareness of human behavior, as embryonic as it is, exists …

"*The P2P Network Is The USER" is definitely a part of our very near future.*