Application design on the Internet

David Clark

6.978J/ESD.68J

Lecture 2 v1.0

Spring 2006

Application design

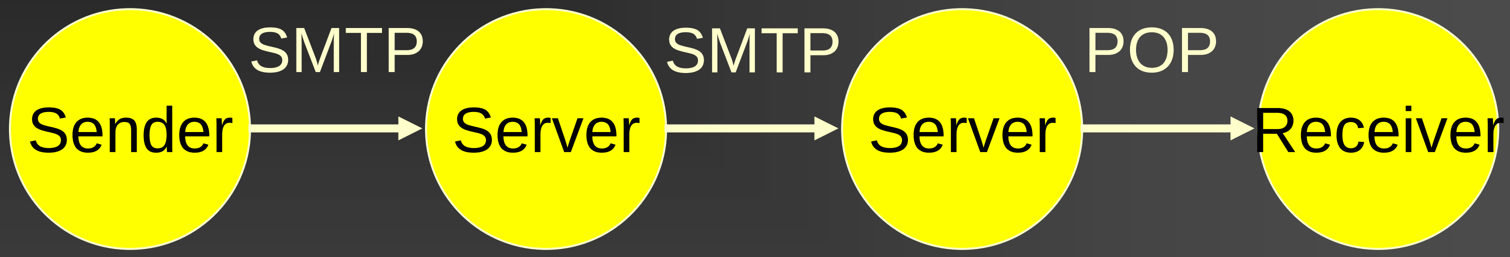
- Remember: applications run “on” the Internet. They are not the Internet.
- Many approaches to construction
  - Patterns of communication.
  - Use of end node software and server software.
- Design considerations such as:
  - Ease of use and deployment, robust operation, low cost, simplicity, lawsuit-proofing, easy management.

Where to start? Servers

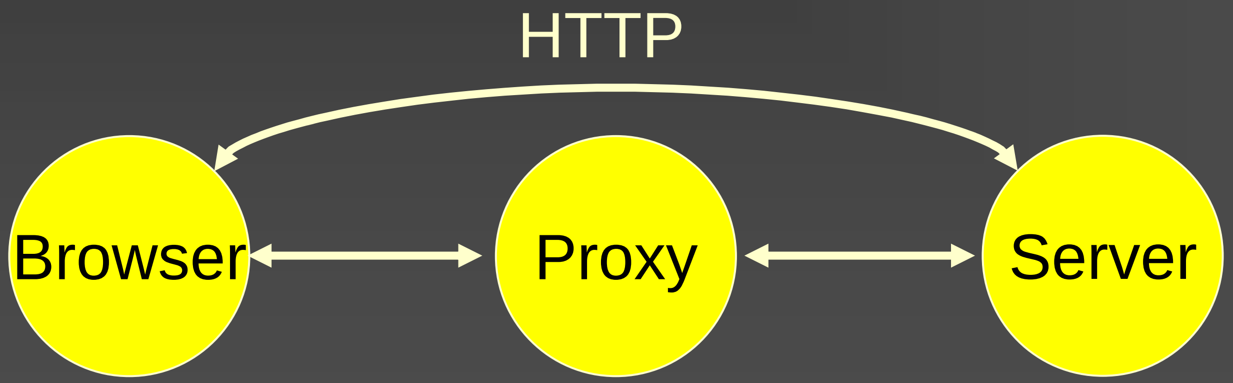
- Modern apps do not follow a simple end to end model.
- (End to end at application level)
  - Remember the end to end argument?
- They are full of servers and services run by third parties.

Some examples:

Email:

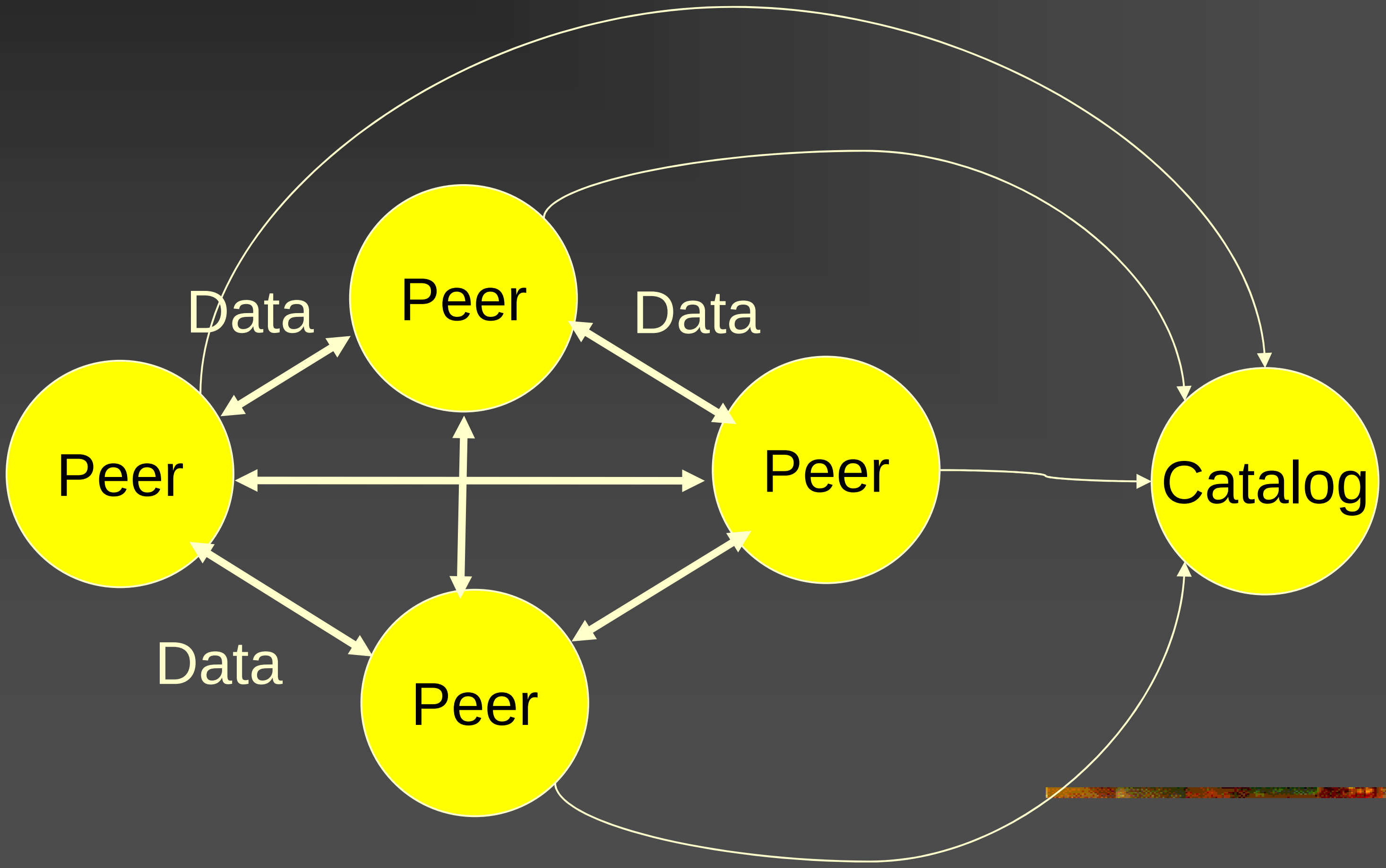


Web:



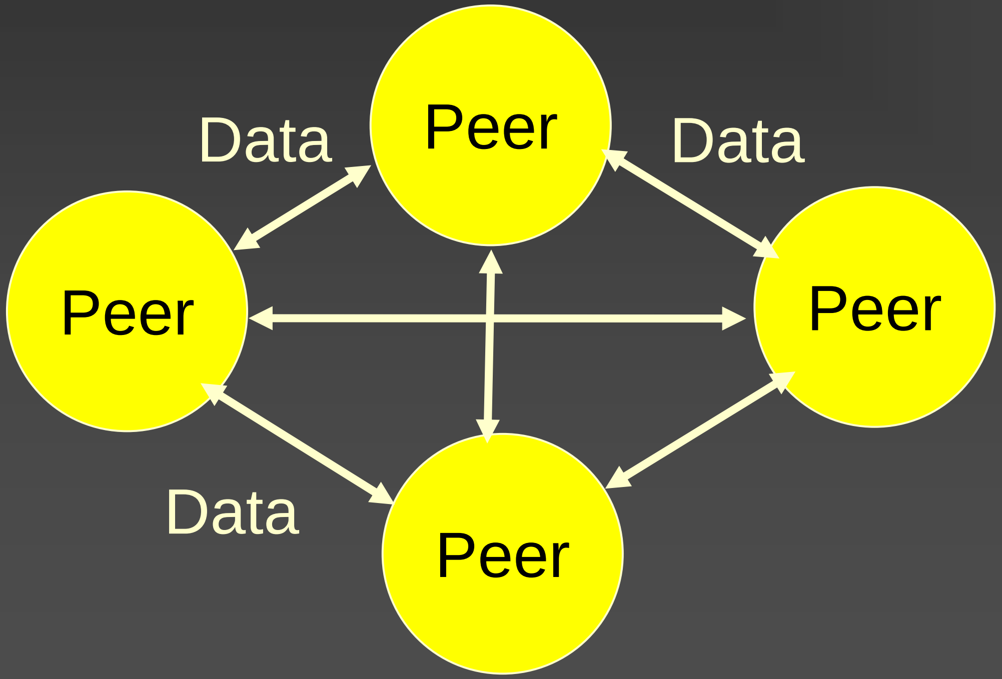
More examples:

Napster (early peer to peer)



More examples:

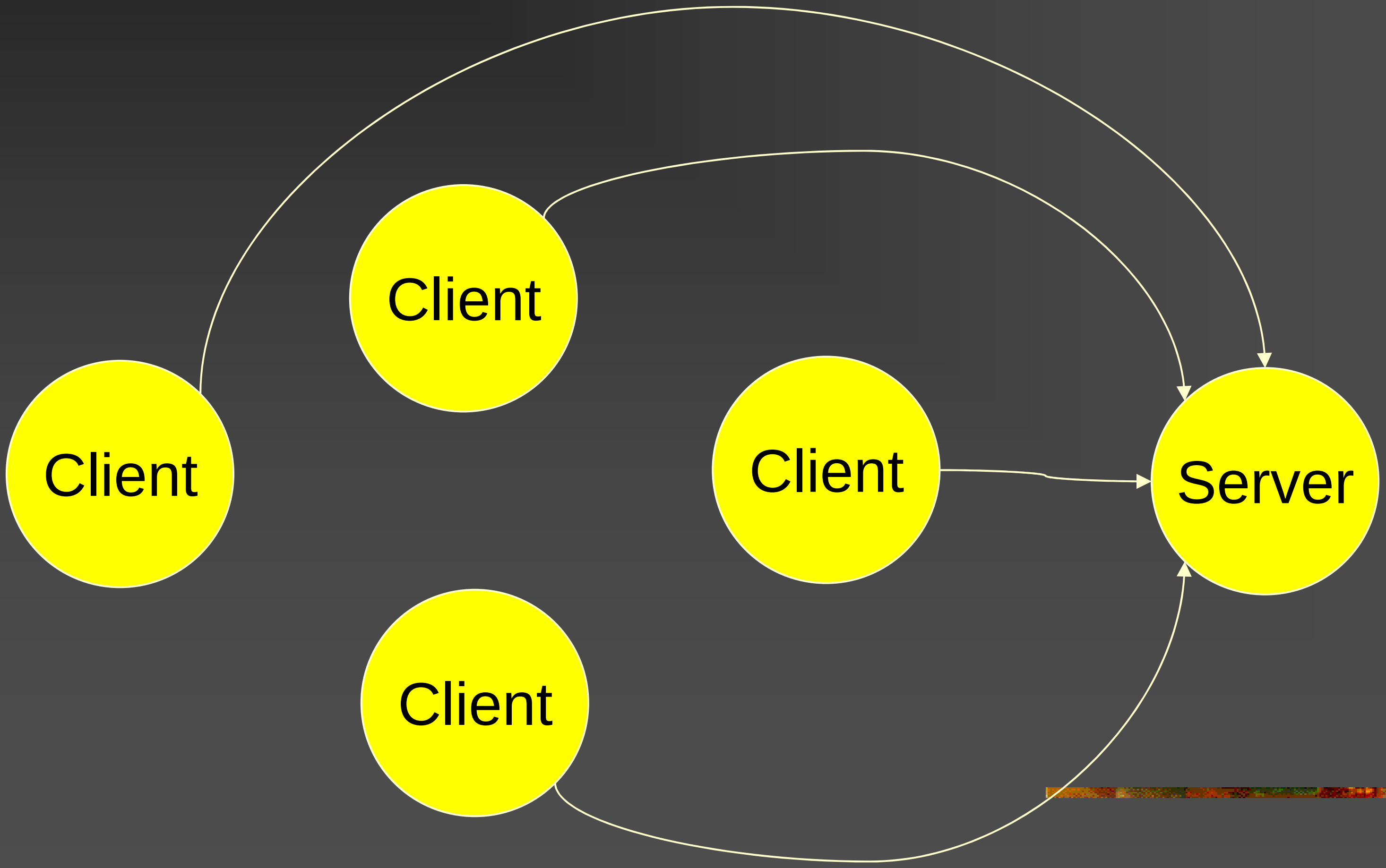
Later peer to peer



Just go feel around for the data. It might be there somewhere.

More examples:

Games (some), and IM (some)



Lots of reasons for servers

- Stage content close
- Pre-process content
  - Specialized device
  - Constrain actions
  - Filter content
- Replicate functions for robustness
- Manage identity
  - Centralize authentication
  - Control release of attributes
  - Preserve anonymity
- Outsource functions
- Cope with NAT and addressing issues

Make comms asynchronous

Move between end-nodes

What problems are we solving?

- Ease of use
- Ease of deployment
- Performance
- Economic (industry) structure
- Robustness
- Security
- Who is in control?

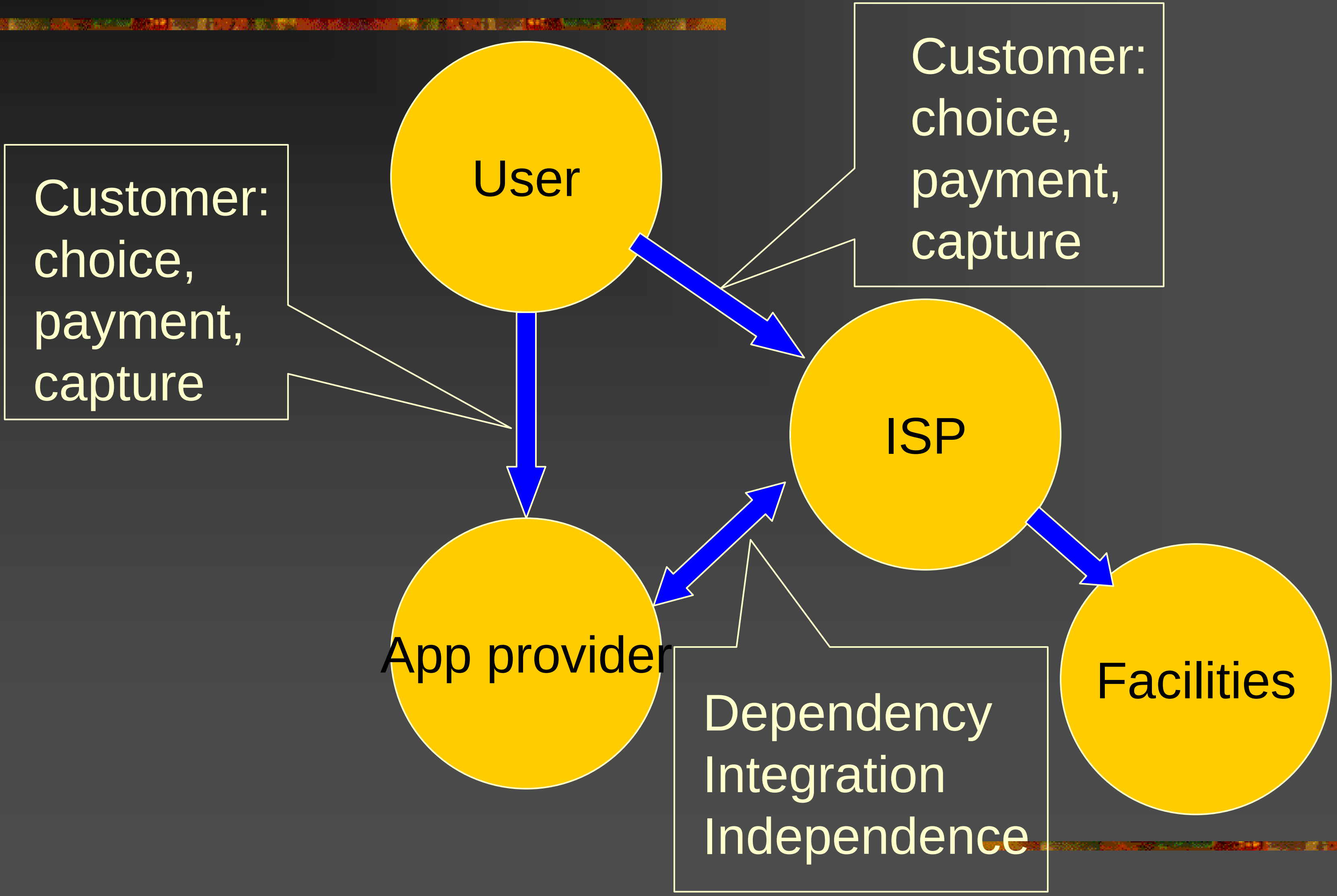
Balance of power

- Who can select the servers to be used?
  - User: delegation (ease of use), filtering (security), pre-formatting, control anonymity, replication and location, protection (applies to both ends)
  - ISP: filtering (value strat, usage control, agent of state), revenue generation
  - Third party (state or “other”): filtering (law enforcement and censorship), monitoring (law enforcement, taxation)

Finding a framework for analysis

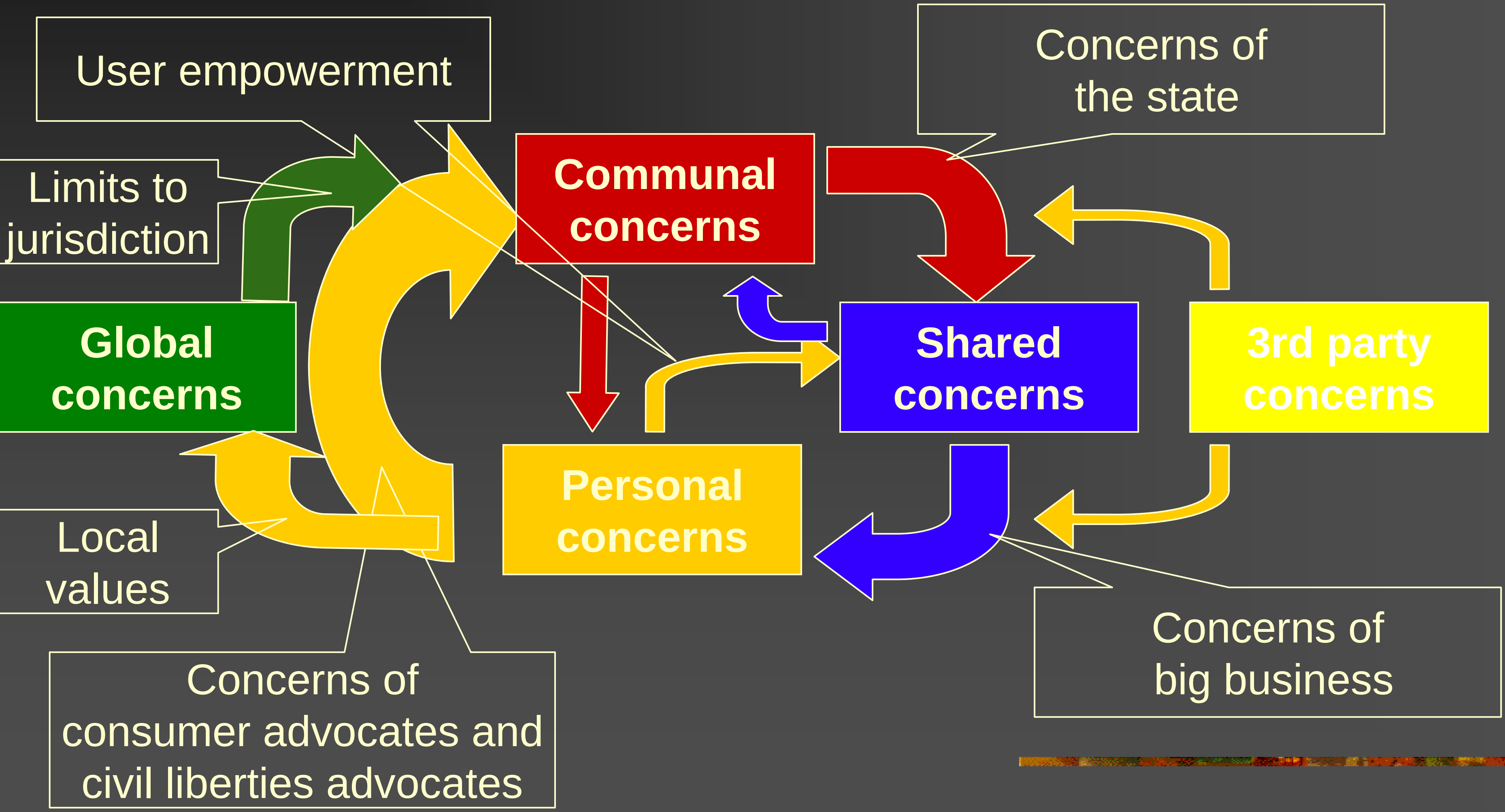
- Need to construct a list of stakeholders
- Need to catalog their interests.
- Two pictures:
  - A value chain stakeholder diagram
  - A broader societal stakeholder diagram

Value chain assessment



Customer: choice, payment, capture
User
Customer: choice, payment, capture
ISP
App provider
Facilities
Dependency
Integration
Independence

A larger space of stakeholders



User empowerment

Limits to jurisdiction

Global concerns

Local values

Concerns of consumer advocates and civil liberties advocates

Communal concerns

Personal concerns

Concerns of the state

Shared concerns

3rd party concerns

Concerns of big business

Some stakeholder concerns

- Users want choice
  - Provides competitive pressure
- Providers want stickiness
- Providers want value stratification
- Providers want to sell services
- ISPs want to balance usage with revenues
- Governments want taxation
- Nations make money on settlement
- Rights-holders want protection

This is very abstract…

- A server is a tangible element.
- It may play many purposes
- The design constrains which purposes it can play.
- The deployment determines what actually happens.

Some topics for consideration

- Relation of app design and addressing.
- The importance of “design for economics”
- An application-centered view of security
- How do applications get deployed?

Topic 1--global addressing

- Thesis: proper design of applications can mitigate the “NAT problem”, and allow us to architect private edge address spaces.
- Use host-specific ports (e.g. RFC 2782).
  - Requires extensive tussle analysis.
- Use servers in global space for rendezvous.

Tussle analysis

- Ports would be random, not well-known.
  - Requires host-specific knowledge to filter.
  - ISP filtering (value strat, traffic engineering) much harder.
  - Firewall filtering much harder.
  - Port scans less useful.
- Name server can be anywhere. Per-service name.
  - Hard to launch location-based attack or scan on name server.
  - But: what names show in messages? What history?
- Example of interaction between application design and network design.
  - A real design question for today. Not hypothetical.

Topic 2--economics

- Should applications be designed taking into account the economic goals of the various stakeholders?
    - User choice of server and provider.
        - Drives competition and controls prices
        - Prevent ISP capture
    - Server-based services are basis for revenue generation.
        - Akamai as source-driven example.
        - Email as receiver-based example.
- Should we concern ourselves with advertising?

Economics of overlay networks

- Overlays are:
  - A tool for sophisticated applications.
  - A tussle tool with ISPs
- If the latter, who pays for them?
  - Cannot scale for free. (Can they?)
- When overlays grow up, who will run them?
  - Option 1: they will be run *by* the ISPs.
    - The major source of new ISP revenues.
  - Option 2: third party providers
    - What will make one option or another the answer?

Topic 3: Security architecture

- Does an application-centered view help sort out issues? (I vote yes…)
  - Re-factor security using tussle analysis.
  - Freedom from attack:
    - Users, end-servers,third-party infrastructure
  - Trusting users who want privacy
  - Untrusting users who want help
  - Third parties that want to intervene
- Who trusts the servers, and how does app constrain this (shared keys, revealed info, etc.)

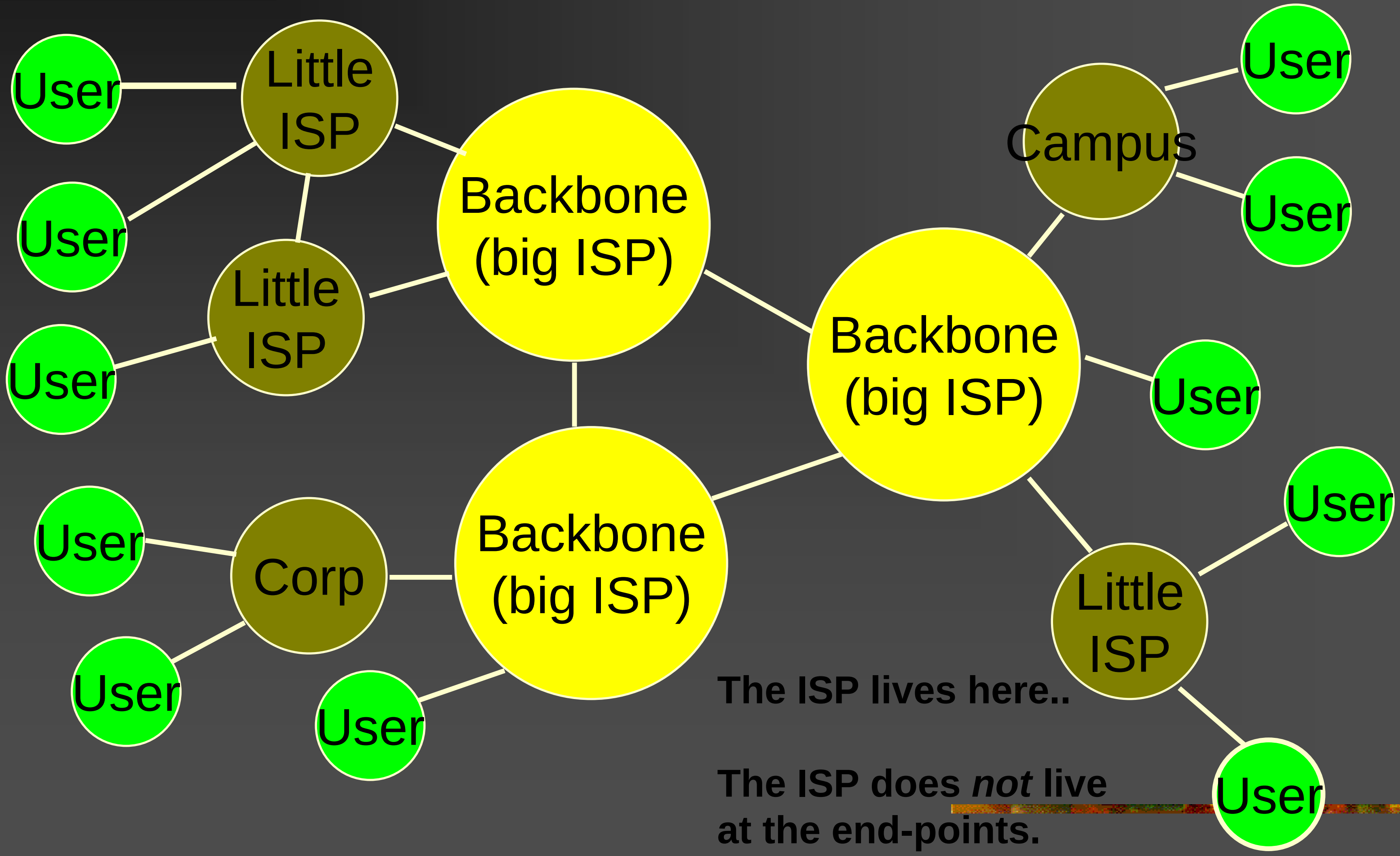
Design rules for secure apps

- Think about:
  - App-level DOS attacks
  - The role of identity and attributes in the above
  - Defense in depth and “outsourcing”.
  - Don’t be an attack amplifier.
- What can be done cross-app? (support service?)
  - Common services vs. app-specific services in the net.

Topic 4: Ease of deployment

- The life cycle of an app, or how do apps grow up?
  - In the beginning, must be end to end. No servers.
  - If successful, lots of folks get interested, and jump in. Leads to servers.
  - In the middle? How about peer to peer?
- The design of an app should take into account how it is to grow up.

A more realistic picture



The ISP lives here..

The ISP does *not* live at the end-points.

ISPs and applications

- ISPs do more than run routers and forward packets.
- They run servers and participate in applications.
  - Each application has its own architecture.
  - This is not the “Internet” design, it is an “application” design.

Questions about applications

- Who controls the servers?
  - Can only the ISP provide them?
  - Can the user select which server to use?
- Is there a topological limitation on the servers?
  - Can my mail server be in China? (Yes…)
- What information has to be revealed to the servers? What is “end to end”?
  - Email--the contents can be encrypted.

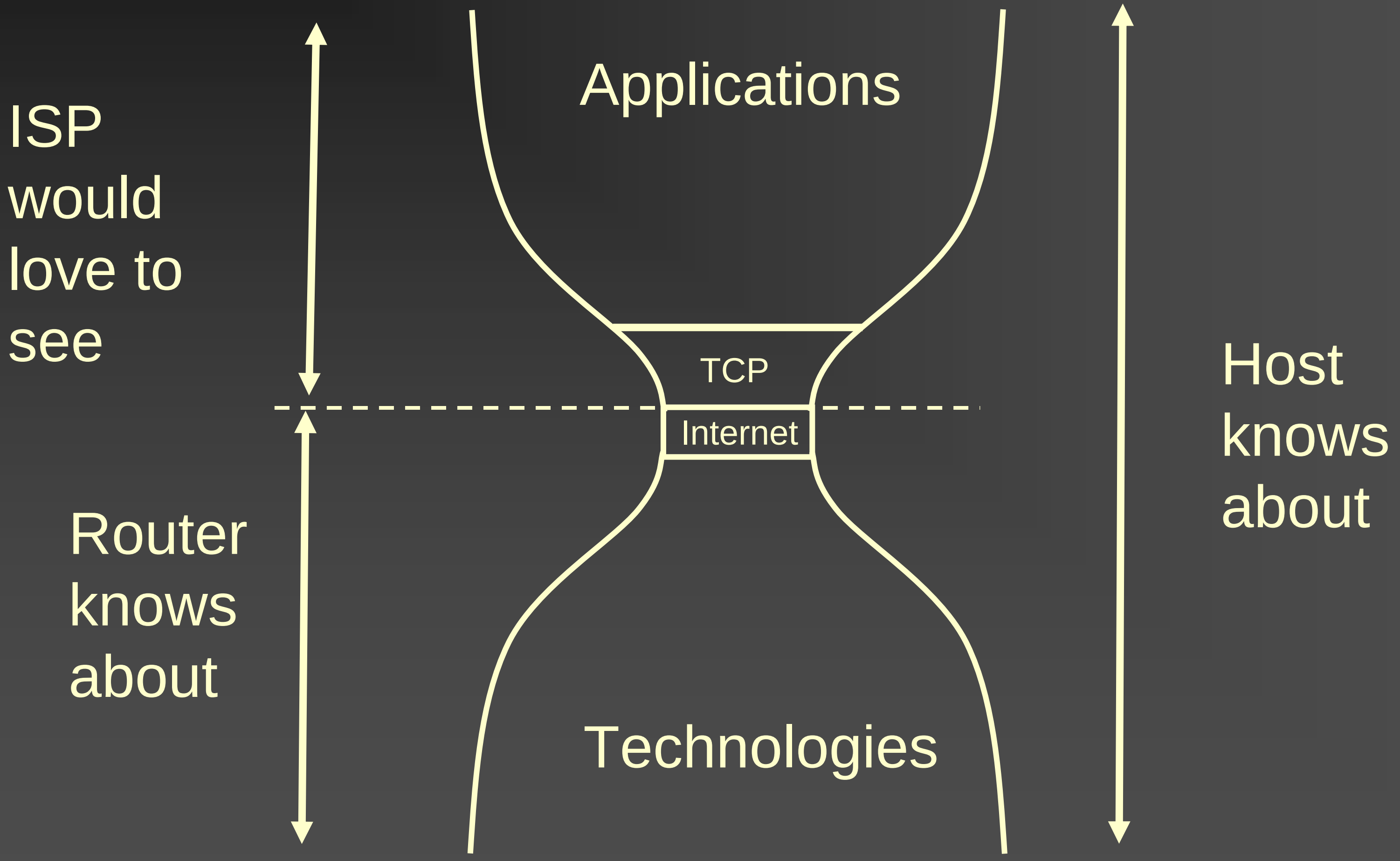
Peeking is irresistible

- What is revealed and who chooses?
- What:
  - What can the servers count, see or change?
  - Email as example of rich design space.
- Who:
  - Sender (mechanically).
  - Receiver (by agreement).
  - ISP (by contract or demand)
  - State (by law)

What has to “show” in the net?

- What transport edges are using?
  - TCP, UDP, RT, etc.
- What application (port) edges are using?
- What Web page is being retrieved.
- Who the parties doing mail or IM are?
- Who the copyright holder on the content is?
- And so on, up the layers.

Host vs. router



ISP would love to see

Router knows about

Host knows about

Peeking is irresistible--so encrypt?