Federation Fosters Freedom

Iljitsch van Beijnum
OHM2013

http://www.muada.com/ohm2013-fff.pdf

Short version

- Network communication can happen in different ways
- Ideally, everyone runs their own server with their own data
- Or at least users can choose from different service providers
- This gives us freedom!

To come

- Introduction
- Case studies: email, IM, P2P file sharing, social networking
- Centralization issues
- What is a protocol designer to do?
- The future: federated search?
- Q&A

Introduction

Designing protocols

- Doing it is (fairly) easy
  - get some data, push it through the network
- Doing it ***well*** is ***hard***
  - spam, authentication, privacy, scalability, speed, efficiency, back/forward compatibility, ...

Some RFCs

- Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming
  J. Arkko, I. van Beijnum, RFC 5534, June 2009
- Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers
  M. Bagnulo, P. Matthews, I. van Beijnum, RFC 6146, April 2011
- DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers
  M. Bagnulo, A. Sullivan, P. Matthews, I. van Beijnum, RFC 6147, April 2011
- An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation
  I. van Beijnum, RFC 6384, October 2011

Terminology

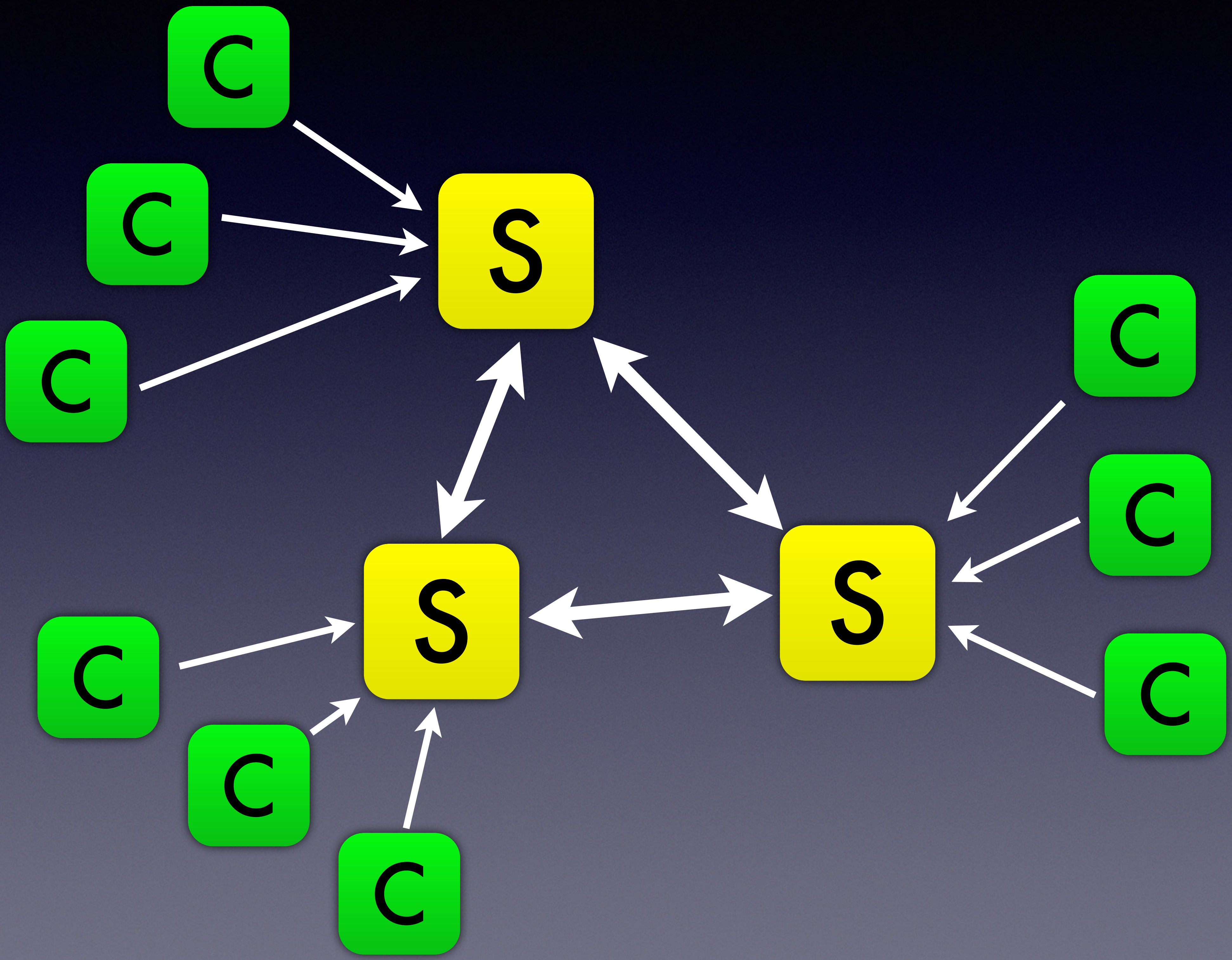
- Centralized: everything goes through a central place
- Decentralized: central coordination, but most things stay local
- Federated: independent, autonomous systems that can, but don't have to, talk to each other

How to communicate

- Network protocols determine how communication happens
    - central design
        - easy to control/intercept
    - distributed/federated design
        - less control, harder to intercept

Case study: email

Email



The protocols

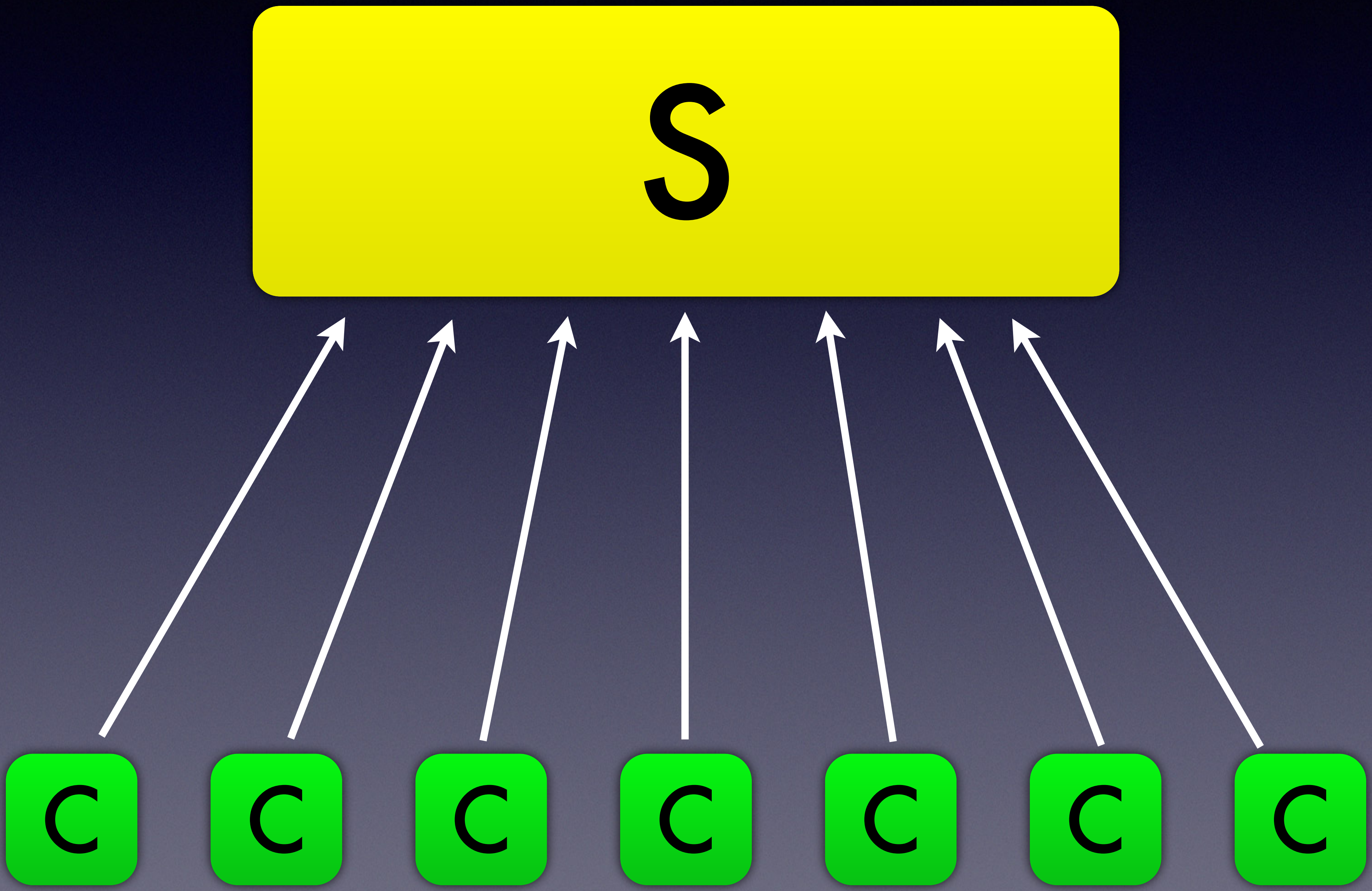
- Email is very old and very simple
- Store-and-forward: submit message to a server, sends it to the next, eventually arrives at the destination
- Federated: everyone runs their own email server, but the servers talk to each other

SPAM!

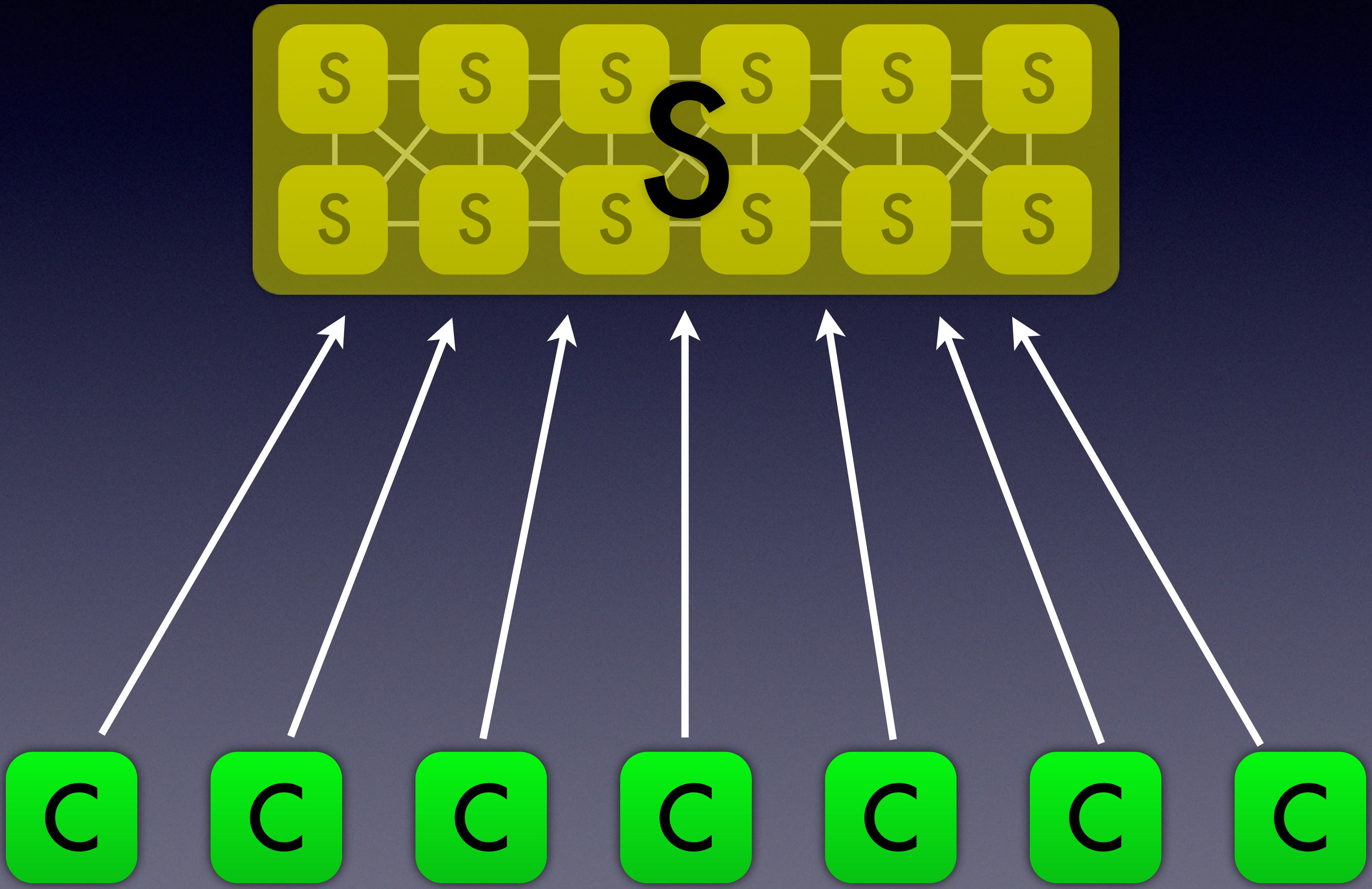
- No authentication
- So can't reject misbehaving users
- Never managed to really solve this later

Case study: instant messaging

ICQ/MSN/AIM



ICQ/MSN/AIM



History of IM

- Early days:
  - talk, ntalk, ytalk, BBS chat
- 1988:
  - Internet Relay Chat (IRC)
- Late 1990s:
  - AIM, ICQ, Yahoo, MSN

IM features

- Since the late 1990s expected features of IM are:
- A buddy list that shows availability
- One-to-one chat
- Group chat
- Audio/video conferencing ability

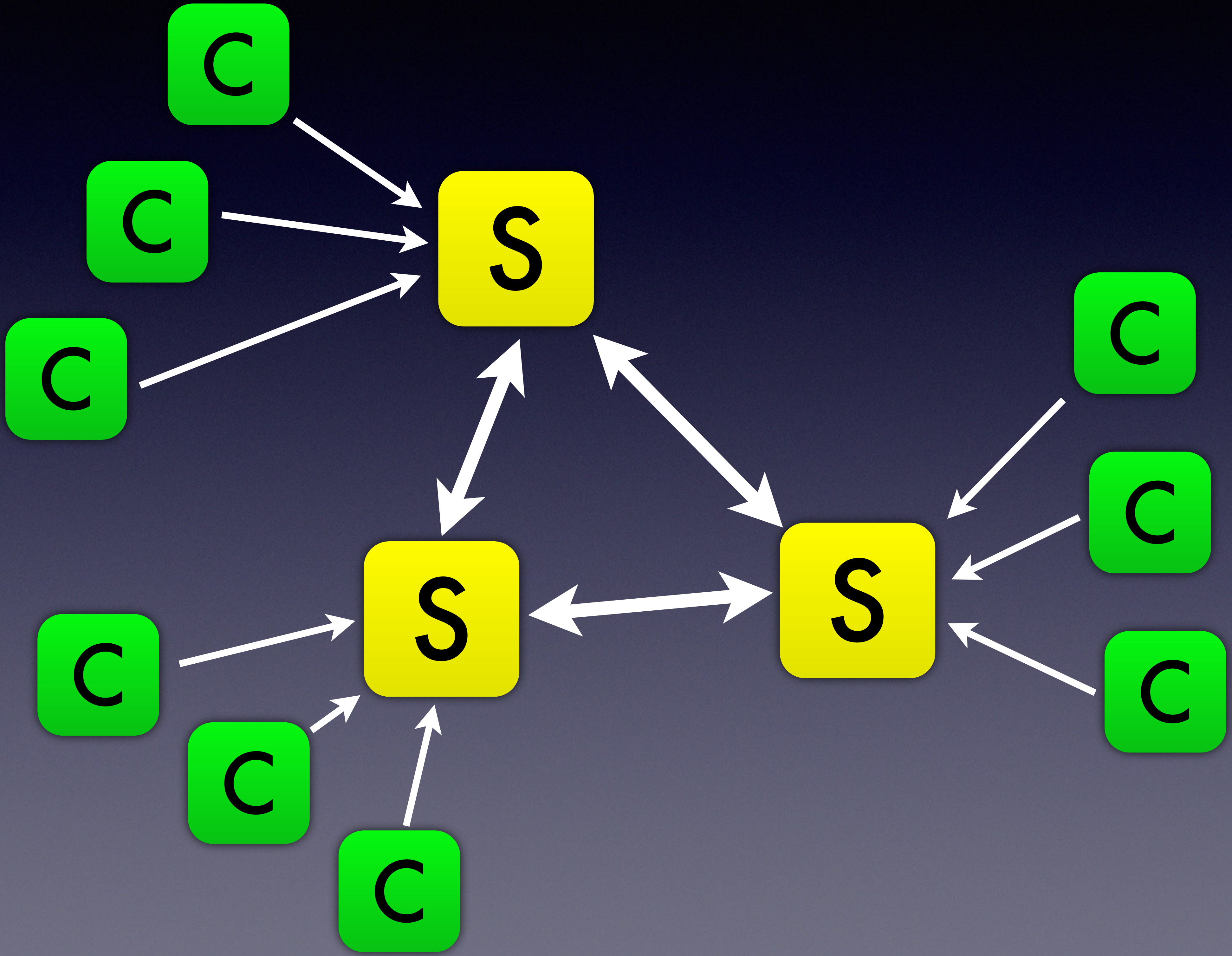
How it works

- Client connects to a server
- Server sends buddy status updates in real time
- Text messages typically flow through the server
- Audio/video bypass the server (for bandwidth/latency reasons)

Jabber/XMPP

- Open alternative to proprietary, non-interoperable IM solutions
  - RFC 6120
- Names/addresses: user@domain
- Domain part identifies server
  - everyone can run their own!

XMPP/Jabber



Open protocol (ab)use

- Google Talk = XMPP
- Skype uses SIP to talk to PSTN gws
- Facebook does XMPP in some way
- Apple uses many open protocols, such as XMPP for iMessage
    - but in a "walled garden":
        - can't XMPP to iMessage users

(about) Facetime

- Steve Jobs, 2010: "We're going to the standards bodies, starting tomorrow, and we're going to make FaceTime an open industry standard."
- That never happened

Necessary, not sufficient

- So decentralized protocols are a necessary condition, but not a sufficient condition
- You can't have a decentralized/ federated service using "jsmit133" type usernames
- But you ***can*** run a closed, centralized service using jsmit@smit.nl type usernames.

Case study:
(illegal) peer-to-peer file sharing

File sharing

- Use an FTP server
- Use a web server
- IRC DCC (direct client-to-client)
- But:
  - bandwidth, too visible (FTP, web)
  - not visible enough (DCC)

Napster

- Everyone makes their local files available
- Download directly from other users' computers (peer-to-peer)
- Central server knows who has what
  - this makes the people running that server liable for illegal use

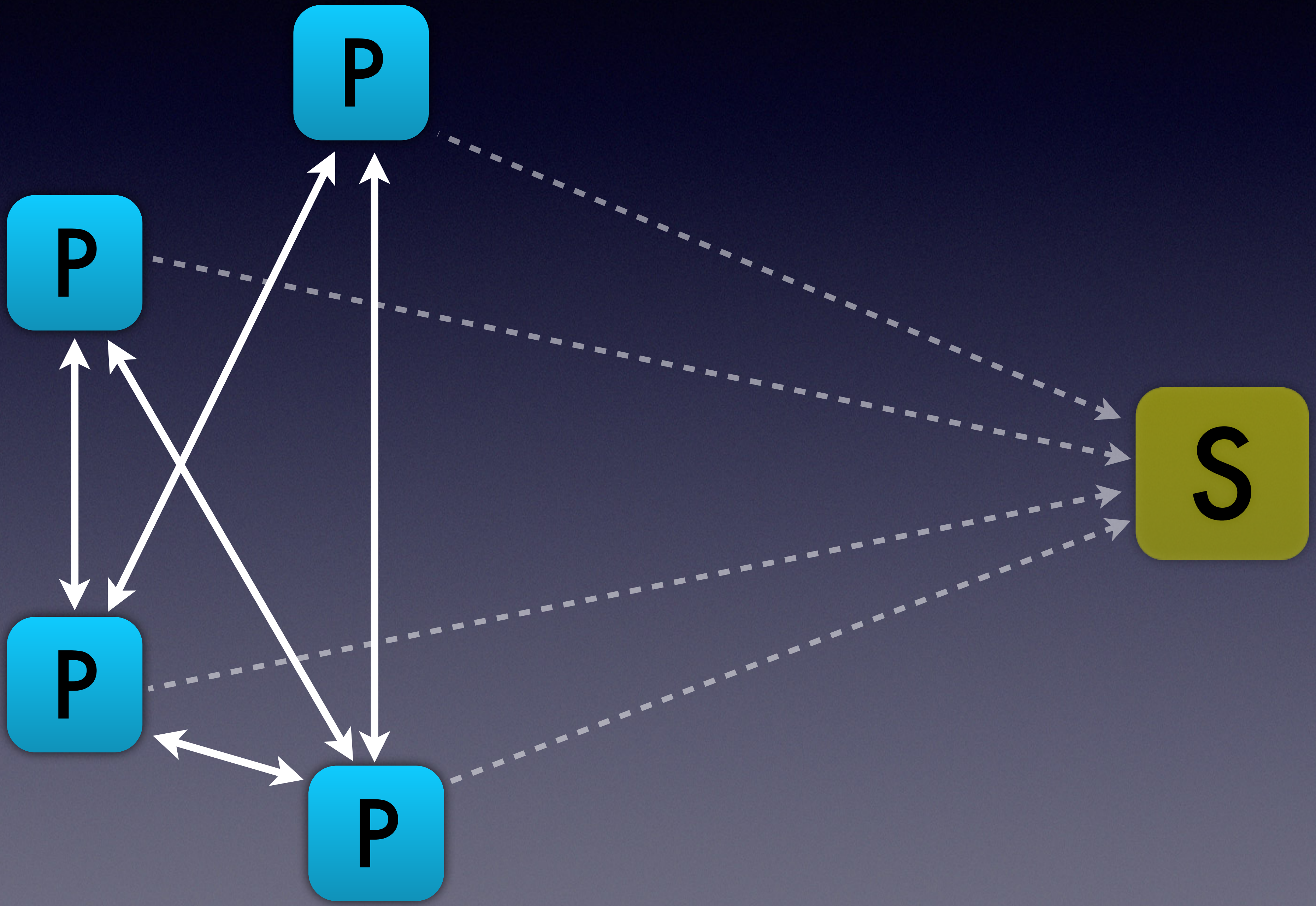
Gnutella

- P2P data transfers like Napster
- But no central database
- Searches are propagated from peer to peer
- No central place to direct legal action against
- But: unreliable/slow searching

BitTorrent

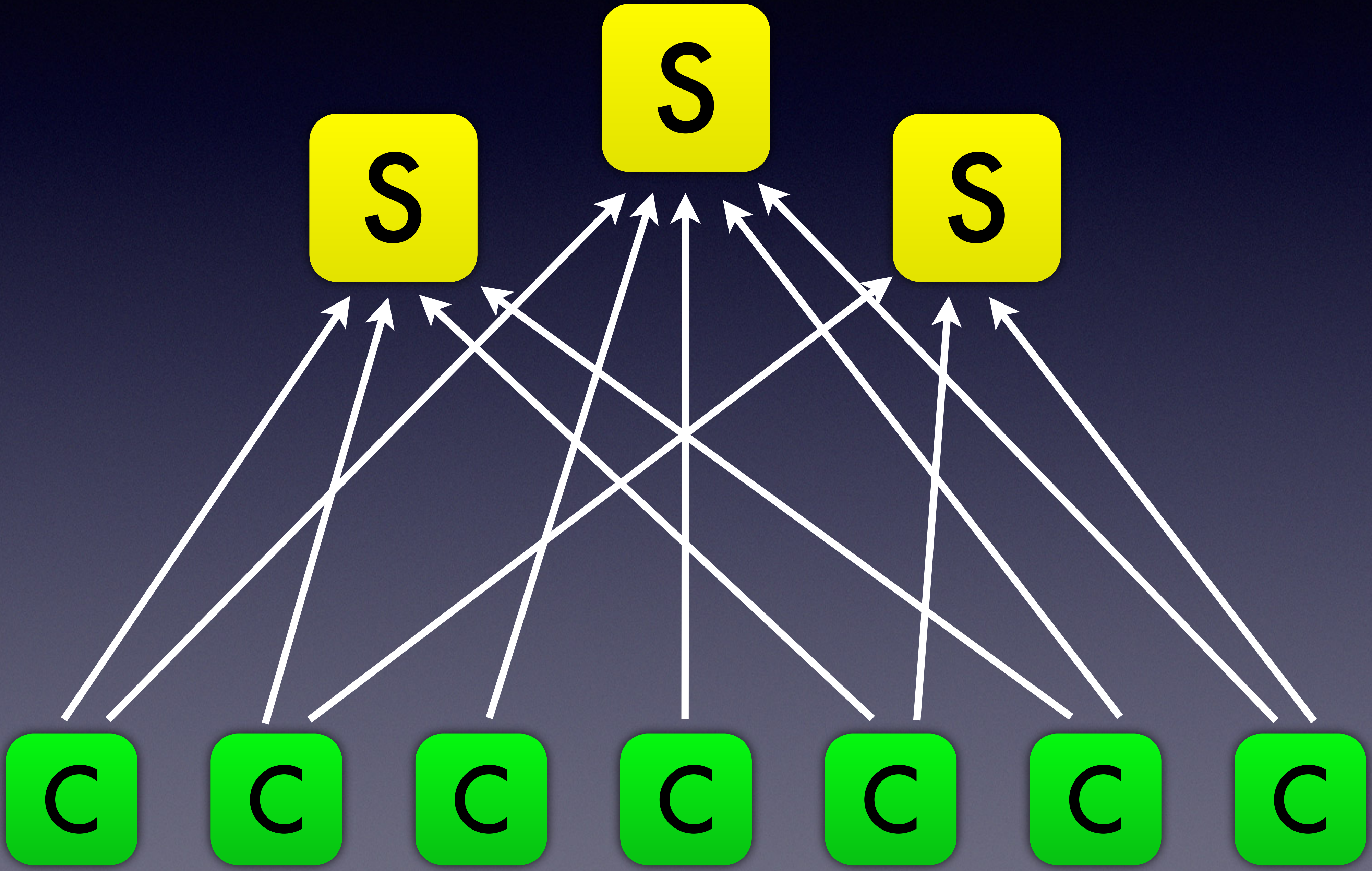
- Rather than download whole files, exchange small parts
    - efficient way to exchange very large (sets of) files
- Originally each transfer coordinated by a central tracker
- But later trackerless, coordination though dynamic hash tables (DHT)

BitTorrent

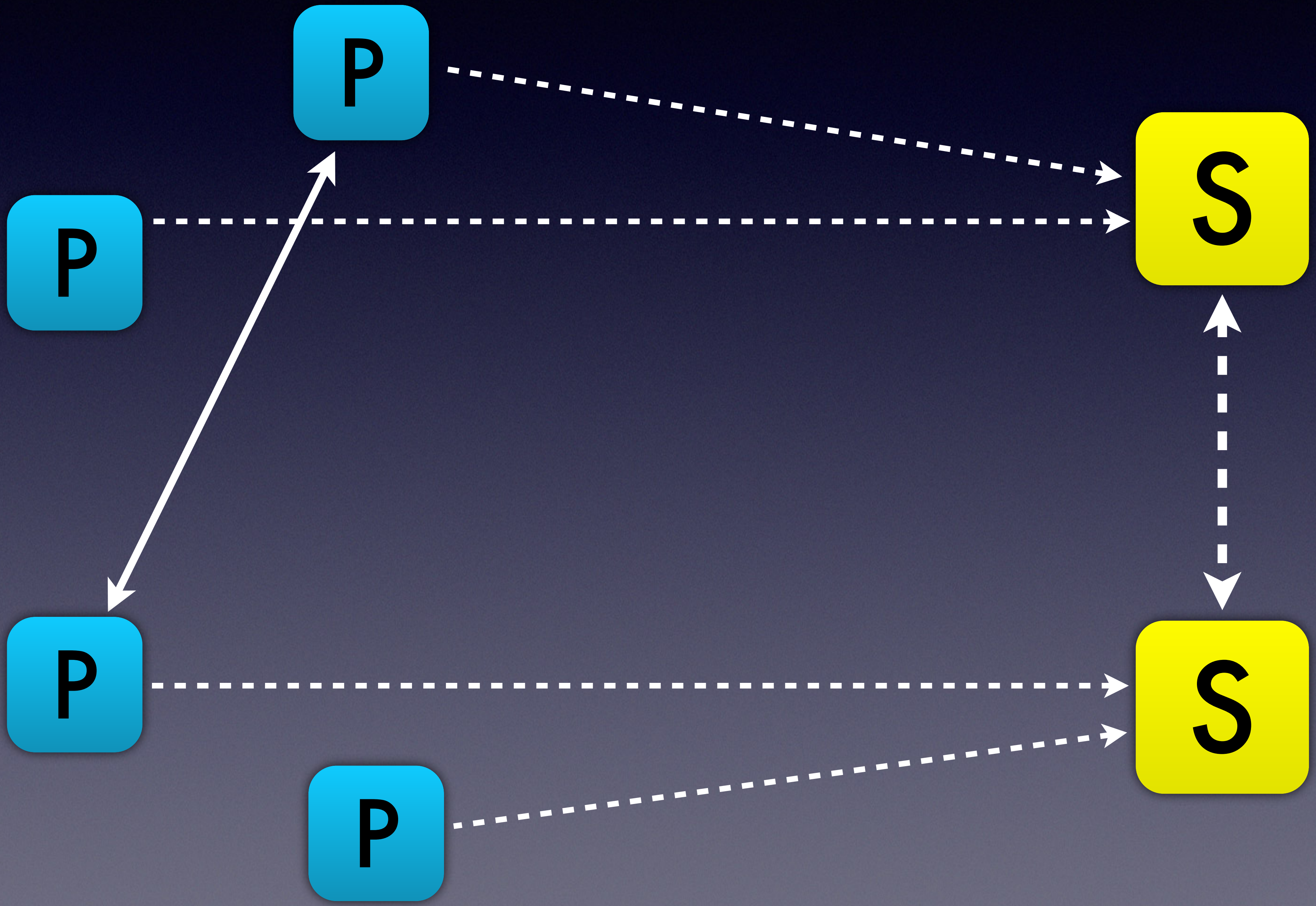


For good measure

The web



SIP (VoIP)

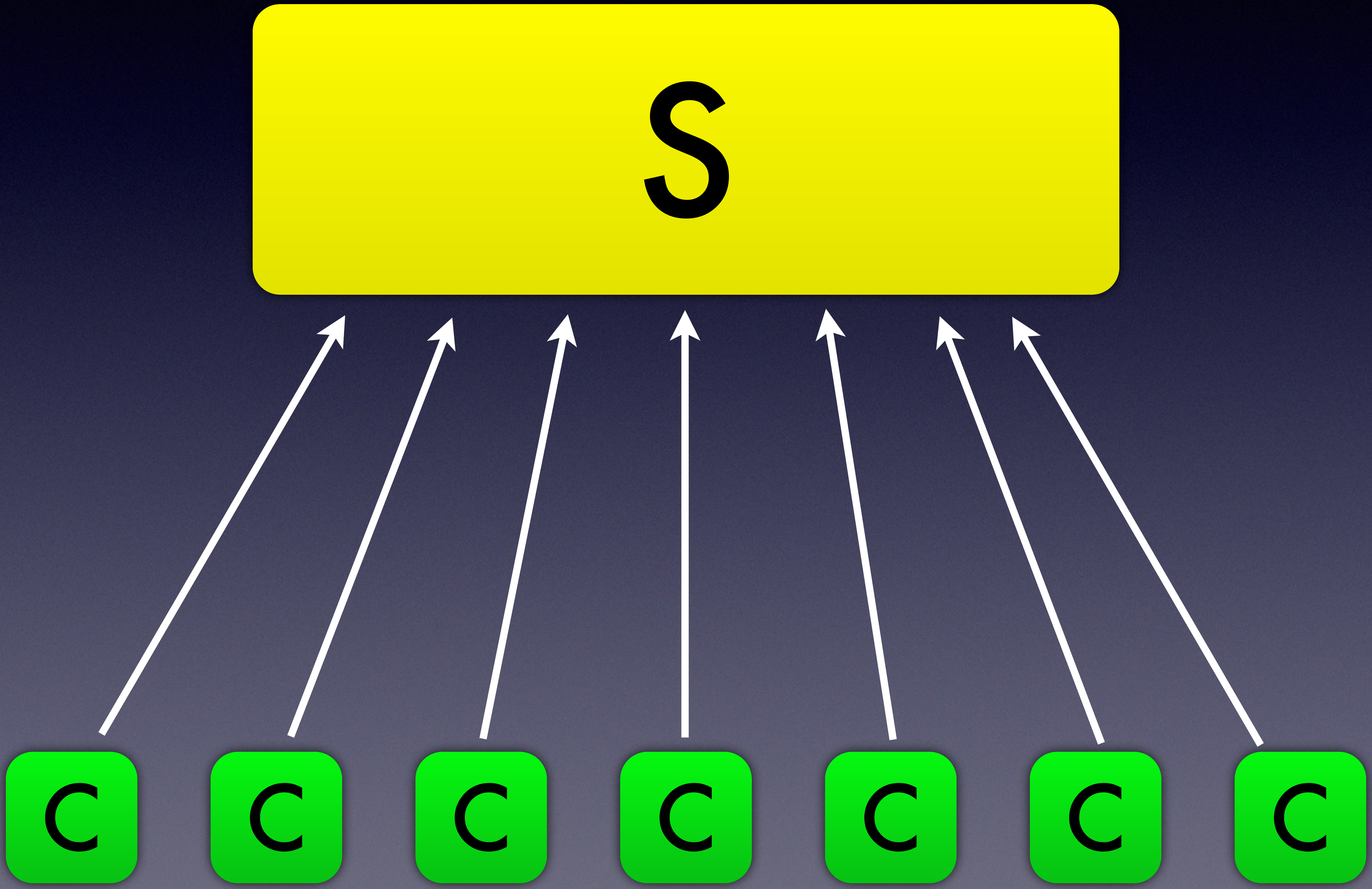


Case study: social networking

SMS with the world

- Crazy idea: what if you can send SMS-style messages to the whole world?
- Even crazier: people love it and it becomes huge!
- Crazier still: companies also love it, #plaster #hashtags #everywhere

Twitter/Facebook



The easy part

- Easy enough to store 140-character messages in a database
- This works well until you have more users than the database can handle
- Now you need to ***scale***

Scalability

- Not about raw speed
- 1 woman creates a baby in 9 months
- 9 women create 9 babies in 9 months
- 9 women don't create 1 baby in 1 month

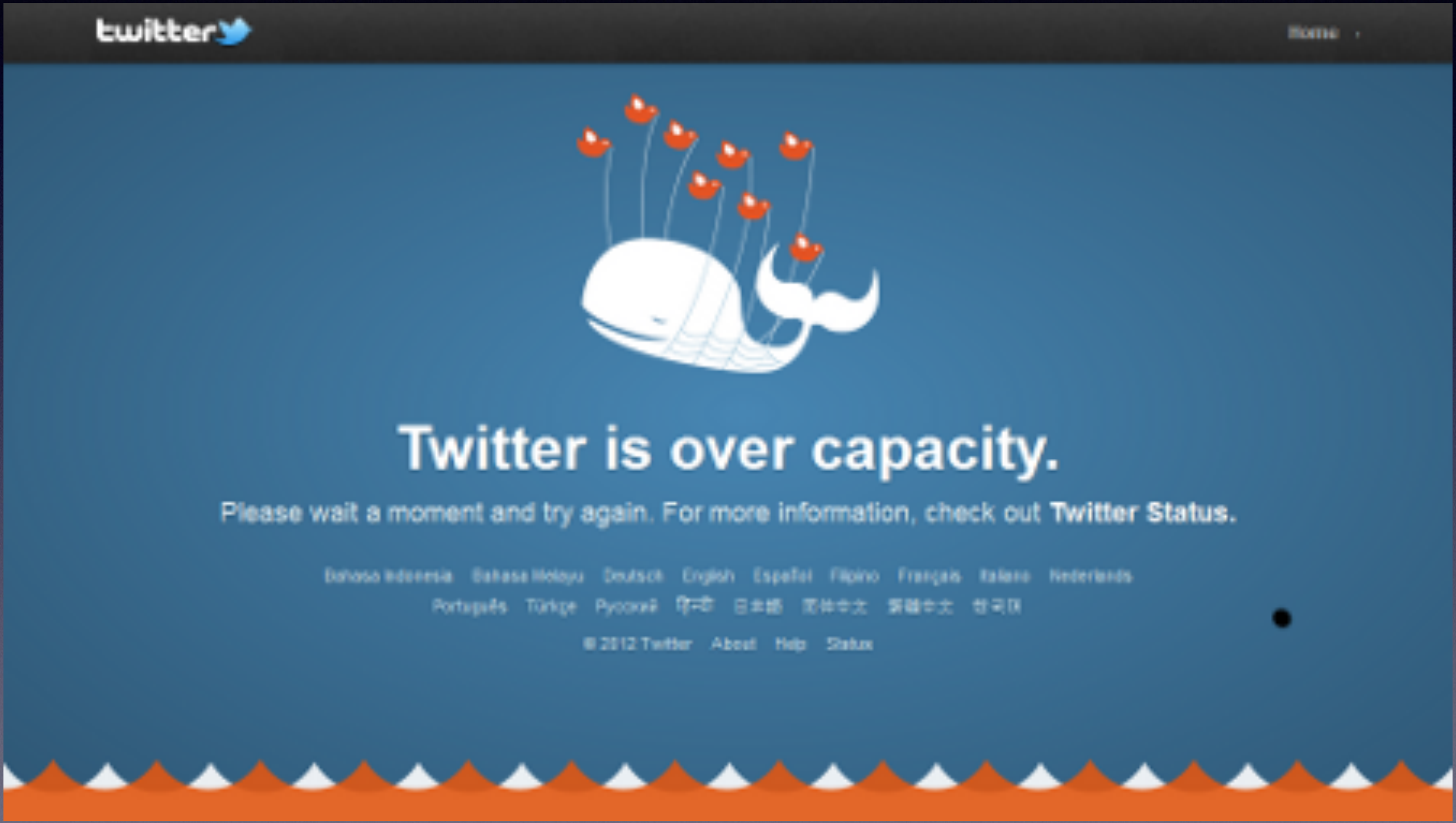
Scalability (2)

- It's easy to do stuff in parallel
  - *if there are no interdependencies*
- Search: my search doesn't depend on yours, can happen in parallel
- Twitter: my feed depends on your update...

Real time

- ...1 second ago. Has to be real time
- Also in the right order
  - (well, except newest-on-top, ugh!)
- Easy when going through a central server
- Not easy without the central server

Early Twitter...


twitter
Home
Bahasa Indonesia Bahasa Melayu Deutsch English Español Filipino Français Italiano Nederlands
Português Türkçe Русский हिन्दी 日本語 简体中文 繁體中文 한국어
© 2012 Twitter About Help Status

Twitter is over capacity.

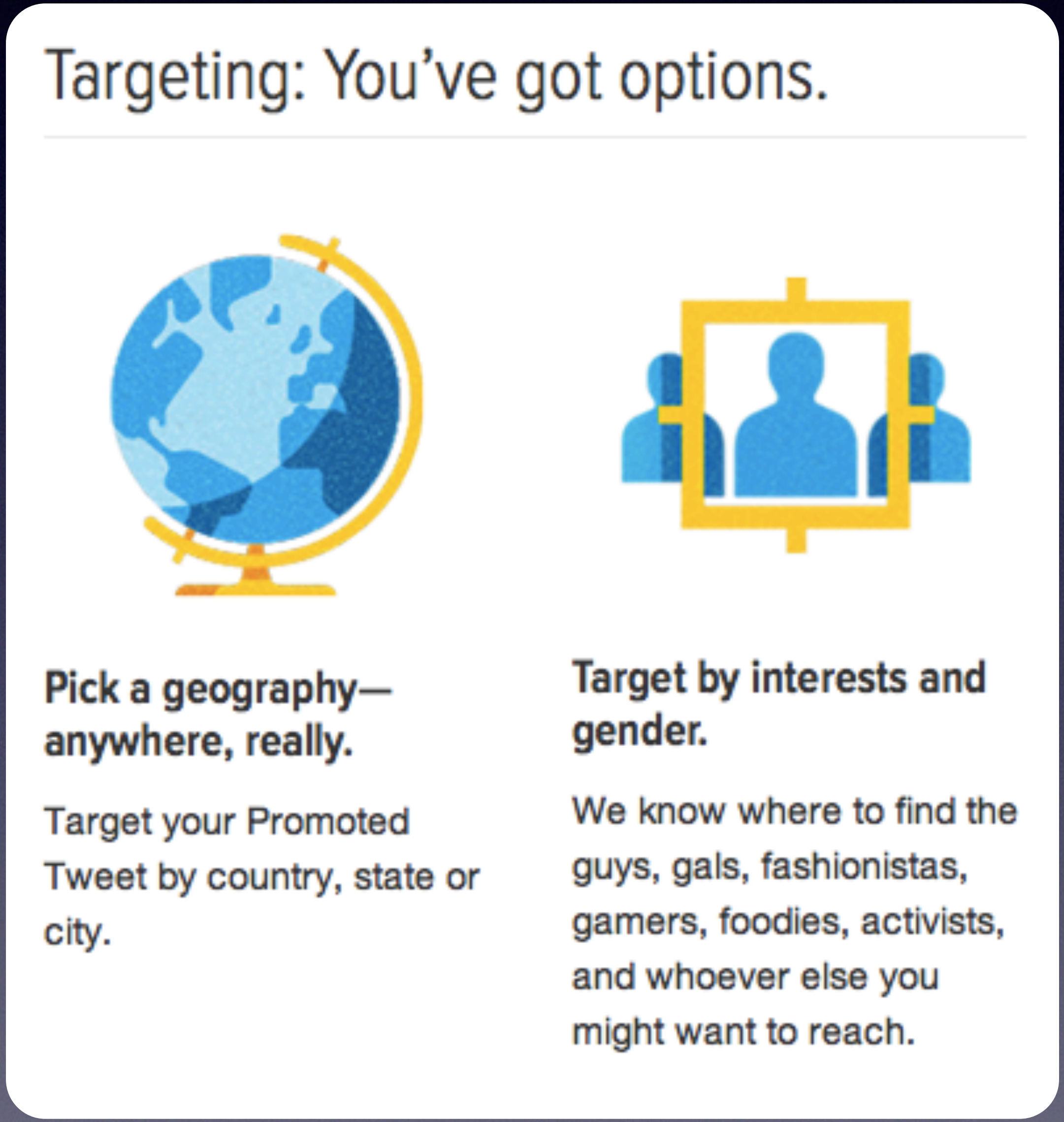
Please wait a moment and try again. For more information, check out **Twitter Status.**

Centralization issues

Gov't and money

I want your data!

S



Targeting: You've got options.

Pick a geography—anywhere, really.

Target your Promoted Tweet by country, state or city.

Target by interests and gender.

We know where to find the guys, gals, fashionistas, gamers, foodies, activists, and whoever else you might want to reach.

Jurisdiction issues

- Servers are likely located in another country
- Where you can't much influence the government and law makers
- And you may have fewer rights as a foreigner than as a resident/citizen
  - (i.e., *unlimited* NSA spying)

Terminology

- Unsolicited commercial messages:
  - in email:
    - spam
  - on Twitter:
    - their business model

Business models

- Way back in 2007 nerds liked Twitter and vice versa
- Grow fast = lots of expenses = lucrative business model = restrictions on clients & APIs, intrusive ads
- Could be worse: Google Reader
- ***One company can kill the service***

The trains run on time

- There are benefits to a dictatorship:
  - much less actual spam on Twitter/ Skype/AIM than in email
  - no (?) malware in Apple app store
  - no supporting old, crappy implementations until the end of time

The bigger picture

- Why is the internet successful but not (so much) X.25 or ATM?
- Why WWW but not WAP or I-mode?
- ***Because nobody is in charge***
  - no gatekeeper = everyone can do their own thing
  - most stuff fails, some gets huge
  - long tail: special needs addressed

Freedom

- Paying for tech specs: not freedom
- NDAs: not freedom
- Forced "family friendliness": not freedom
- Needing a business relationship with A to talk to B: not freedom
- Closed protocols/algorithms: not freedom

Initiatives

- There are initiatives for more openness in social networking, like
  - OpenSocial
  - identi.ca
- But: Metcalfe's law: usefulness of a network = n^2
  - hard to get critical mass of users

What is a ~~techno-hippie~~ protocol designer to do?

Decentral vs federated

- Isn't a decentralized design good enough?
  - Yes, it is better than centralized
  - No, there are still issues
- For instance, the DNS: everyone runs their own server, but only ICANN (+ US gov't?) can decide about .xxx

Federate everything?

- That would be nice
- And extremely hard to do
    - Gnutella and trackerless BitTorrent:
        - much slower and less reliable than Napster and BitTorrent with a tracker

Maybe later

- Hard to imagine how Twitter could have grown fast as a federated system
- Starting as a centralized system can make sense
  - work out the bugs with full control
  - then decentralize (scalability!), standardize, federate

Namespace

- But choose a federation-friendly namespace from the start!
- Yes, you can always add "@aol.com" to all your usernames
- But this is painful and always creates more trouble than you can imagine
    - like: oh wait, gmail is a protected name in the UK!

Namespace (2)

- So use usernames with a domain part from the beginning
  - possibly allow domain part to be hidden in daily use
- Think about authentication and new user creation, these are fundamental to anti-spam measures

The future: federated search?

Search today

- Google, MSN, Yandex, Baidu spider the web
- Go to their websites to search
- They run their proprietary algorithms and give you (hopefully usable) results

Metasearch

- Metaseach engine: takes a user's search term, submits to multiple search engines
- Cooks the results and presents them to the user
- Limited to the search engine's results
- Not good business for the actual search engines

Domain-specific search

- Many domain-specific searchable databases available
    - Internet Movie Database
    - Online shops: Amazon, Bol
- Search is constrained so results are typically better

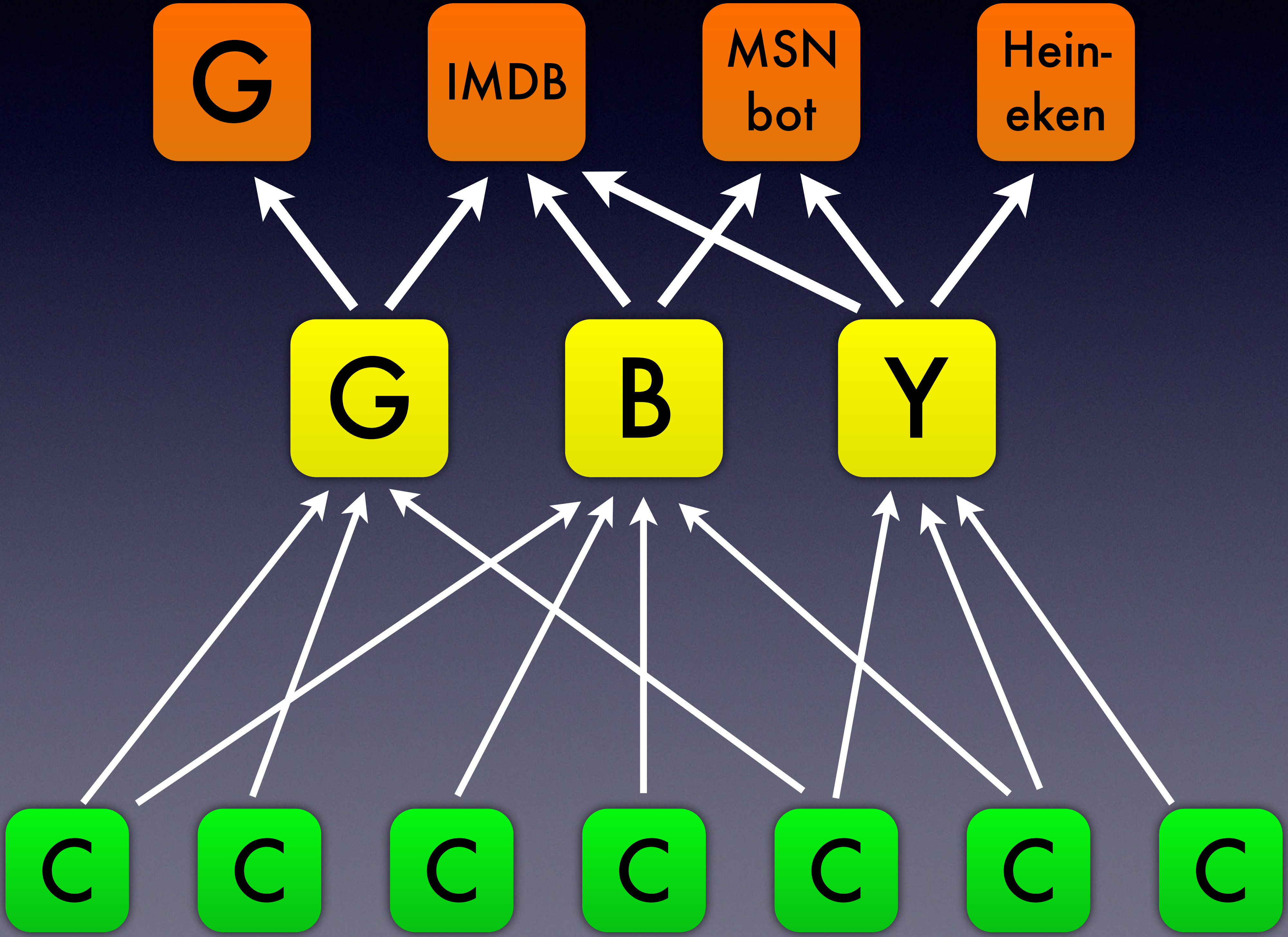
Federated search

- Decouple three stages:
  1. database creation (like spidering)
  2. database querying
  3. results ranking and presentation
- Have different organizations handle 1+2 and 3

Federated search (2)

- So users visit a "search portal" (SP)
- SP sends out search queries to several databases
- Databases return results
- SP filters and ranks the results, shows them to the user

Federated search (3)



Why would this work?

- IMDB has better info about movies than Google
- Heineken probably has better info about beer than Bing
- Competition between databases
- Running a high quality, specialized database becomes attractive

Why would it fail?

- Spam, SPAM, **SPAM!**
- Business model issues for companies running spiders and databases?
  - business relationships databases and SPs may be problematic
- Protocol overhead and waste from duplicated effort

(Good for Google)

- Not automatically bad for current big players such as Google:
- Users won't run away overnight
- They get better access to specialized databases, allowing for higher quality search results
  - (parsing web pages is so crude...)

Questions?

If you think of any later:
http://www.muada.com/
iljitsch@muada.com