Siskin: Leveraging the Browser to Share Web Content in Disconnected Environments

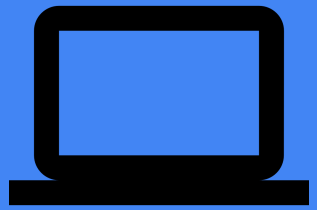
Sam Sudar[1,2], Matt Welsh[1], Richard Anderson[2]
1 Google, Inc
2 University of Washington

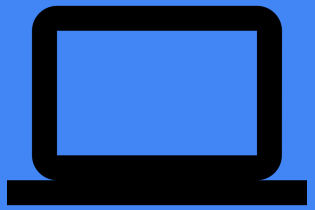
Disclaimer

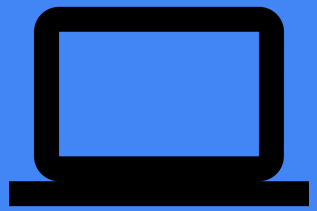
- PhD work from Google internship
- github.com/srsudar/SemCache

There is a huge opportunity for the web to improve education in low-resource settings.

Connectivity in these schools is often poor and intermittent.








Storing the Seeds of Knowledge
egranary
DIGITAL LIBRARY
www.egranary.org

Offline Educational Resources

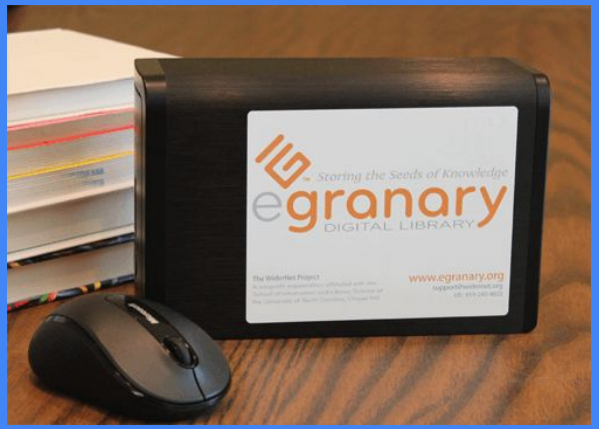


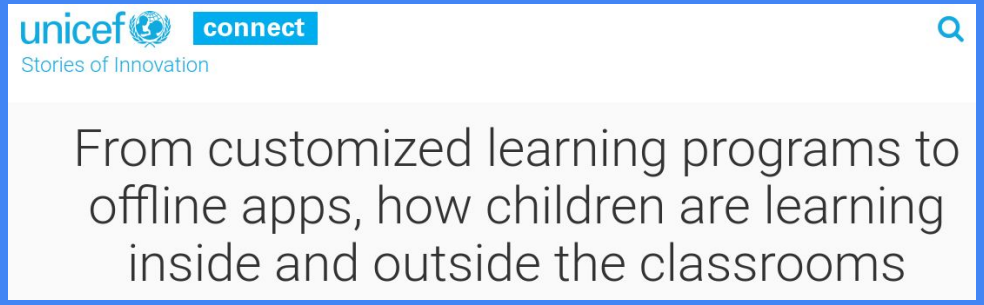
Image Credit: Widernet

Offline educational resources – crucial for African schools in the years to come

Assessing the use of technology and Khan Academy to improve educational outcomes

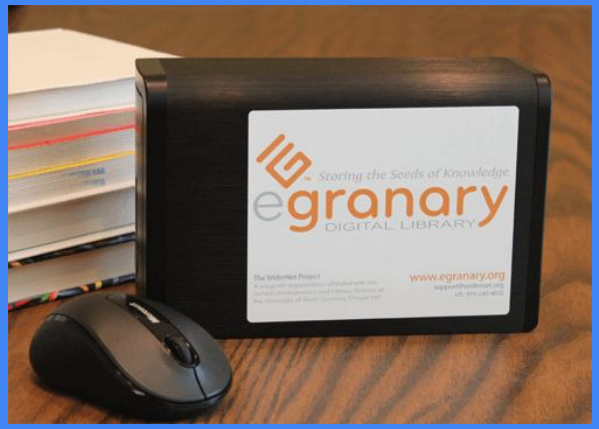
Digital Library Appropriation in the Context of Sub-Saharan Countries: the Case of eGranary Digital Library Implementation

OERs taking Schools from Resource Poor to Resource Rich



Not Perfect

- Can be expensive
  - RACHEL-Pi: $169
  - External USB eGranary: $2,000 + shipping
  - eGranary server: $3,700 + shipping
- Standalone device
  - Updates aren’t built into the equation
  - sysadmin without UI
  - If it breaks you need special training to fix it
  - Single point of failure



Assumptions

- Set of devices on a local WiFi network
- Intermittent or slow connection to the web

Goals

- Allow users to save local copies of webpages
- Discover web content from peers as a distributed cache
- No sysadmin required
- Build as a browser extension so there is no additional hardware

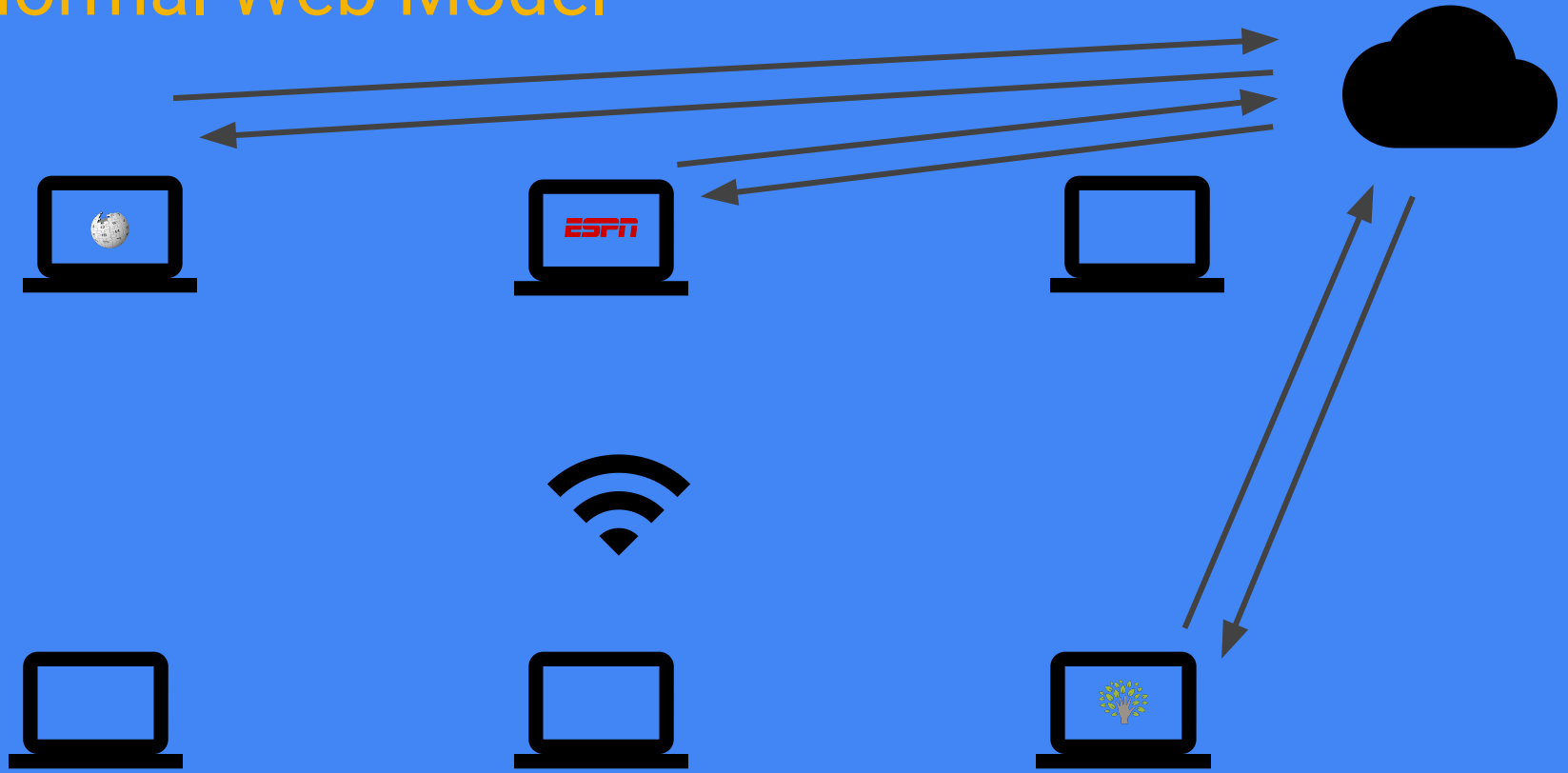
Research Question

Can we leverage web and browser technology to create a sysadmin-free, configuration-free, decentralized web cache to enable offline access to the web in a seamless fashion?

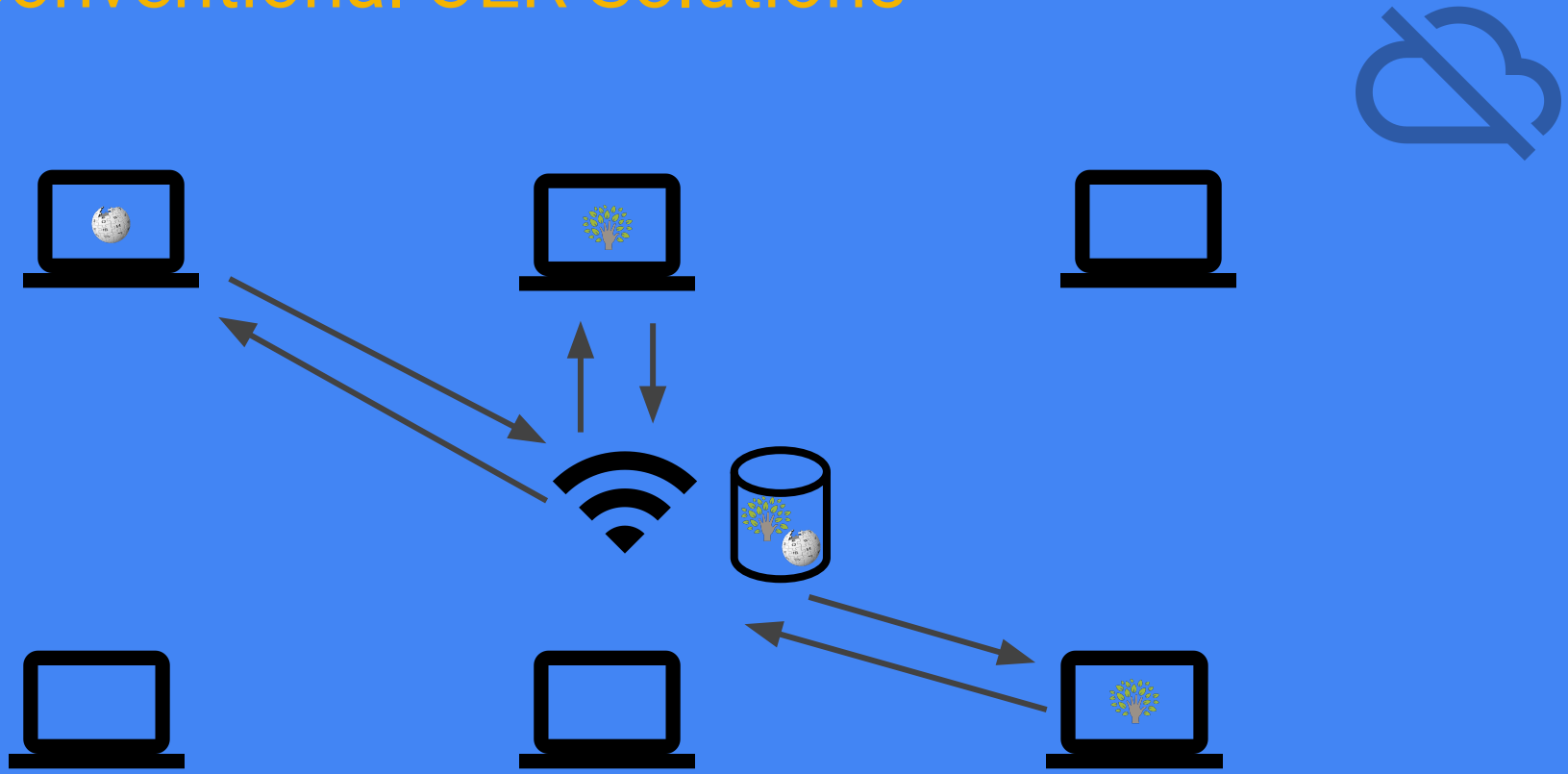
Browser-based approach

- No standalone device
- One-click install model
- Updates automatically (when available)
- No configuration
- No sysadmin
- Built for existing infrastructure
- Decentralized

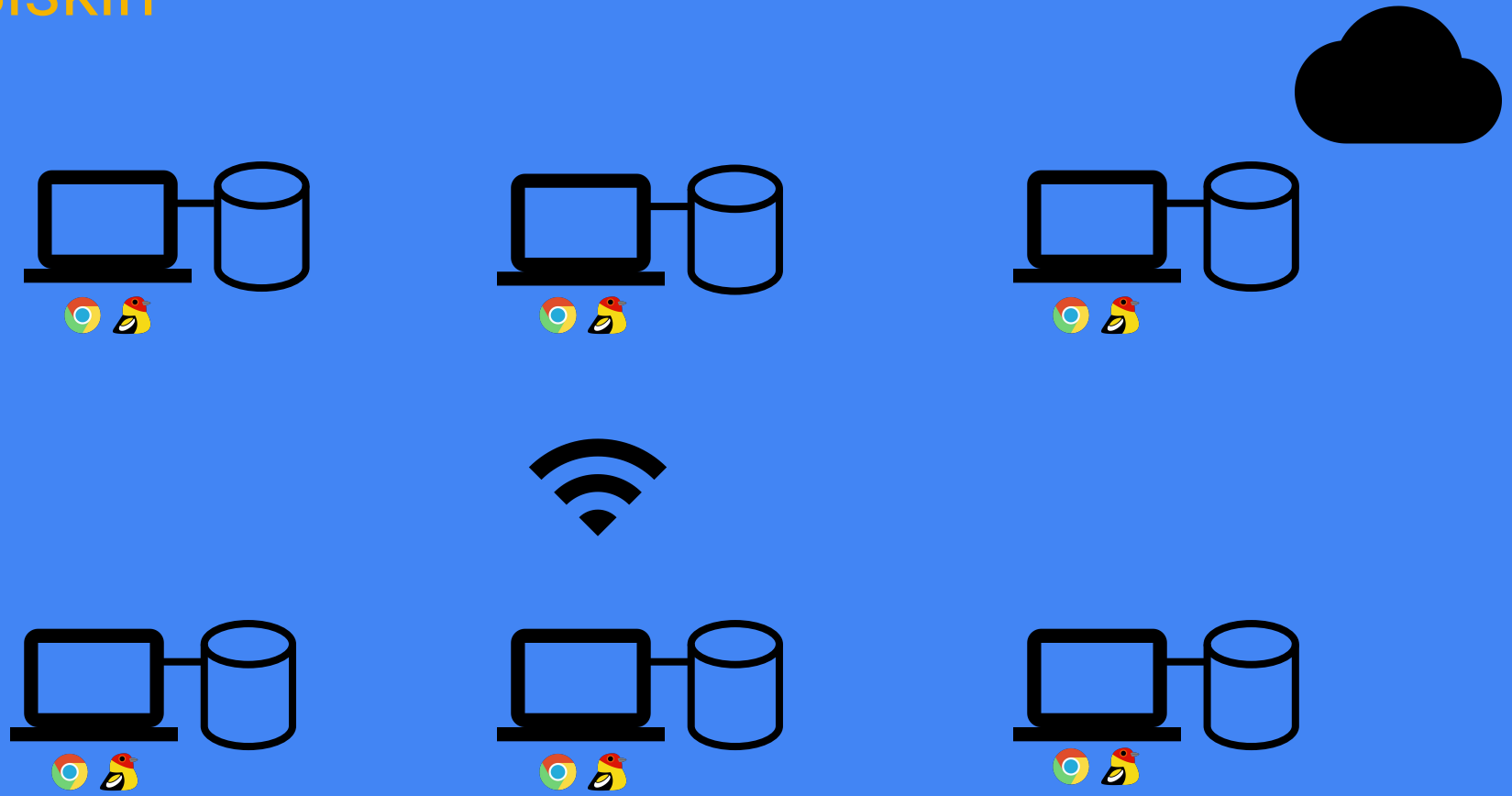
Normal Web Model



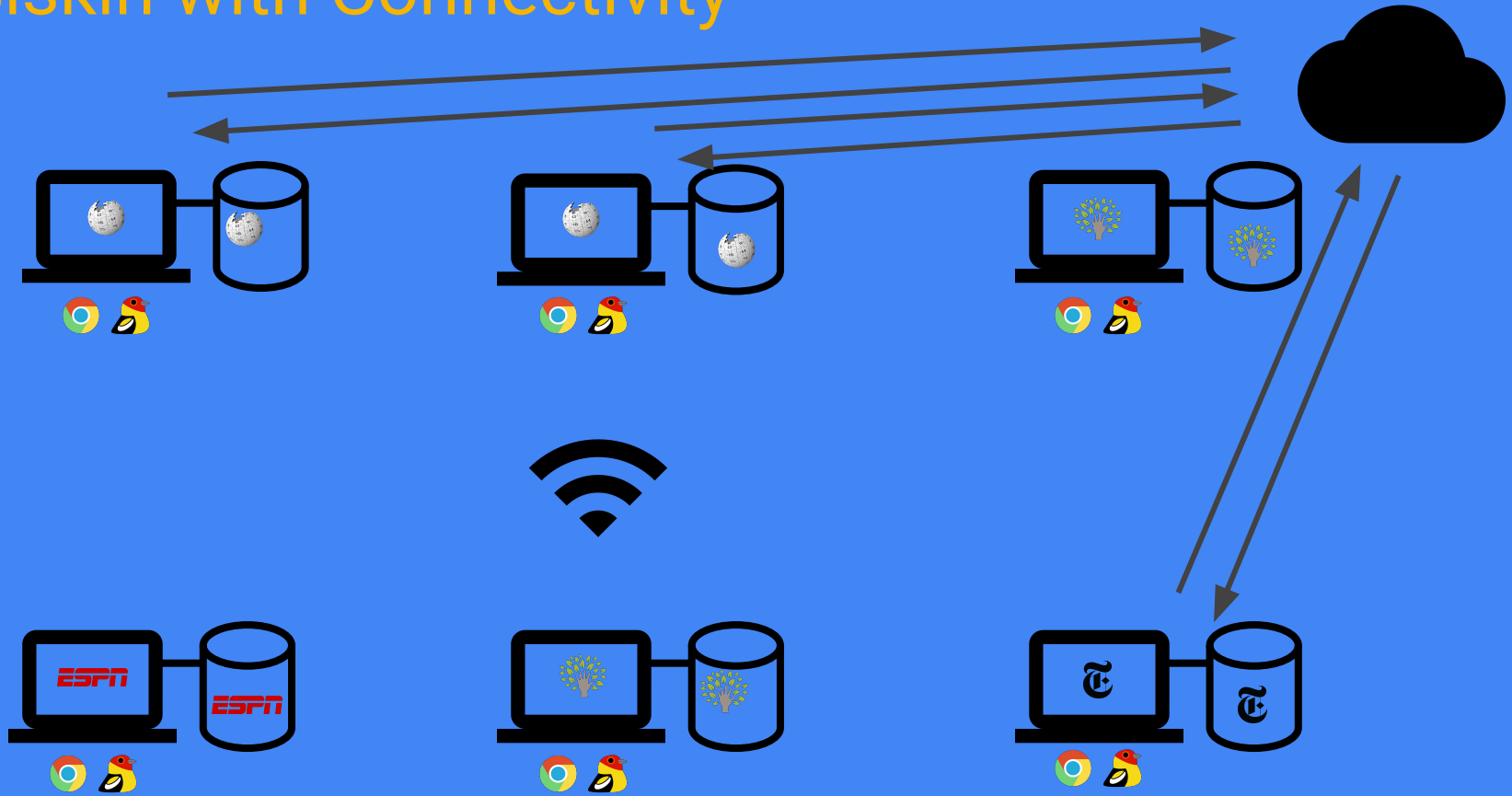
Conventional OER Solutions



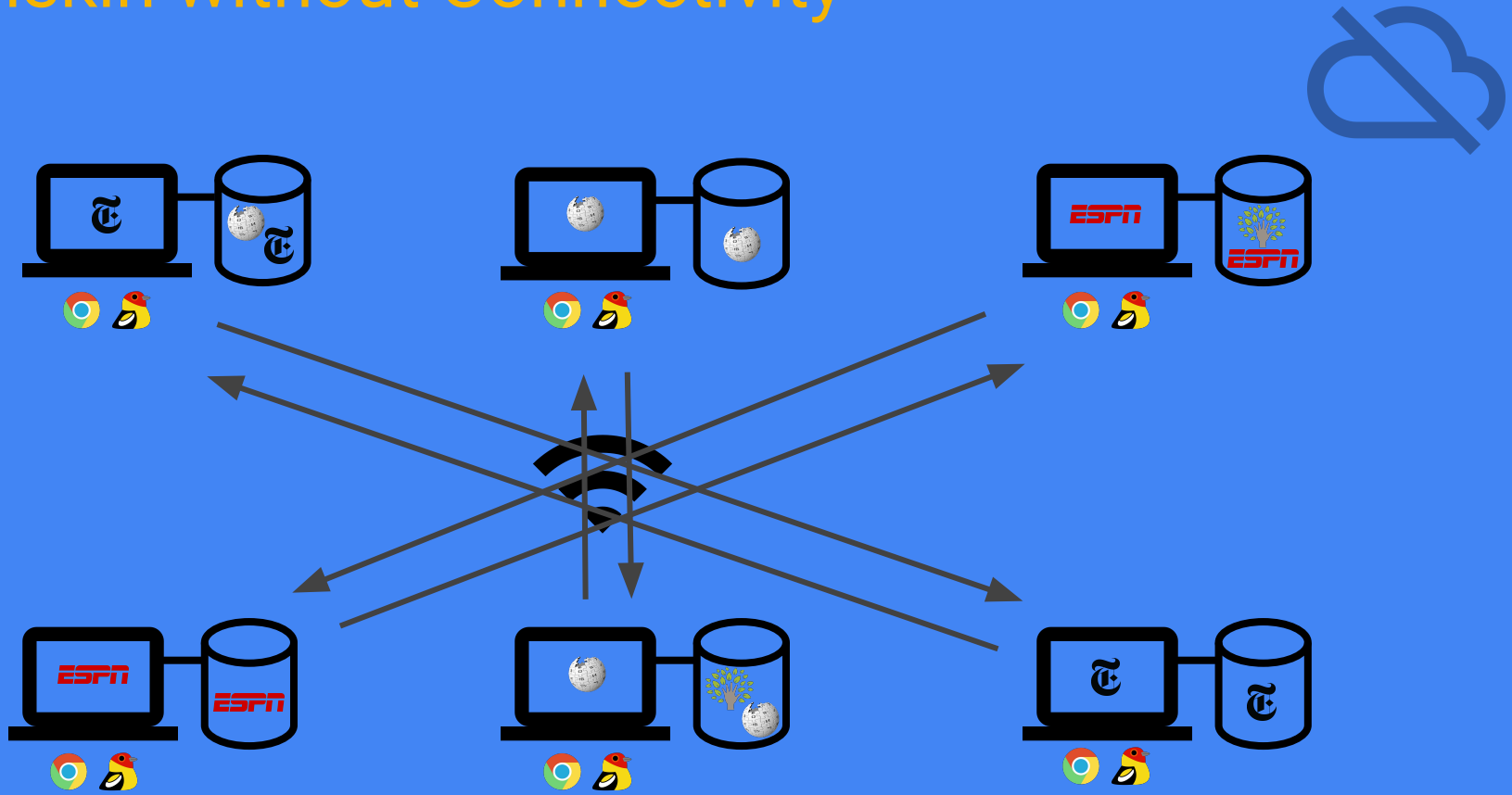
Siskin



Siskin with Connectivity



Siskin without Connectivity



Making Siskin Work

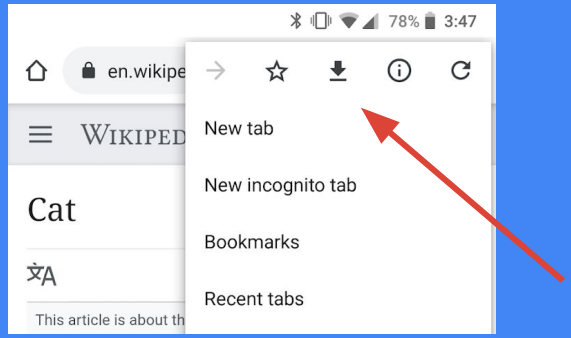
- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer

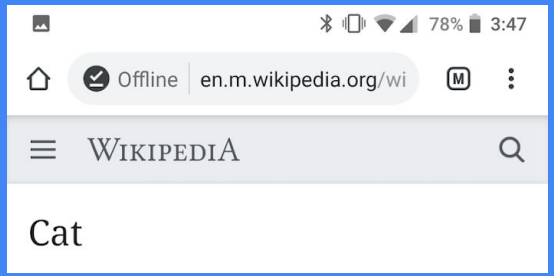
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer

Saving Pages

- MHTML
- Page as single file
- Good, not perfect

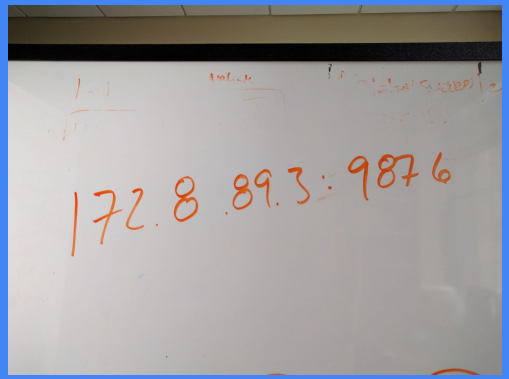




Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer

Naively, how do we find peers?



Zero-conf: ‘it just works’

Multicast DNS

- Network-local DNS

DNS-Based Service Discovery

- Use DNS as a database

Review: Regular IP

172.28.7.80 Goes to one device

Review: Multicast IP `224.0.0.*`

`224.0.0.*` Goes to all devices

Review: DNS

Response comes from remote server

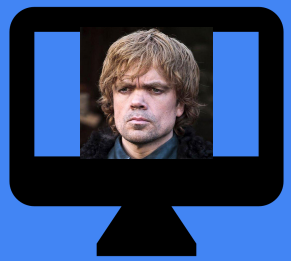
Multicast DNS (mDNS)

224.0.0.251:5353

Response comes from device on local network

DNS-Based Service Discovery (DNS-SD)

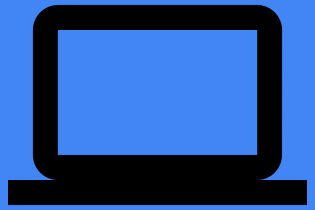
Tyrion



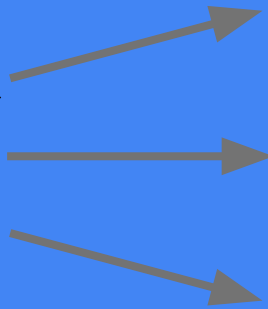
172.28.7.90:8888

DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888



PTR?



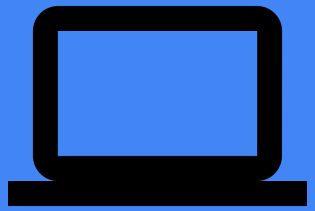
Arya

Jon

Tyrion

DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888



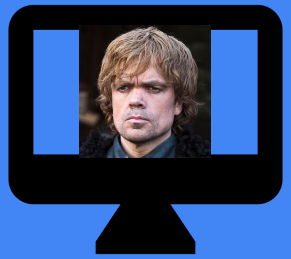
PTR

PTR

Arya

Jon

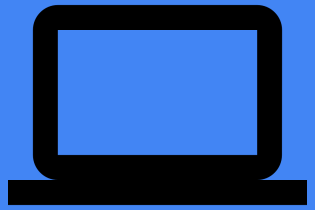
Tyrion



PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp

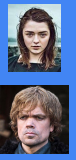
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888



SRV?
Tyrion.siskin.tcp

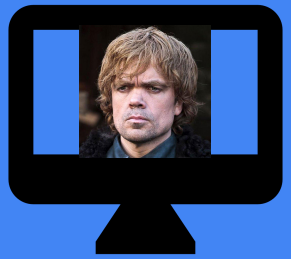
PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp



Arya

Jon

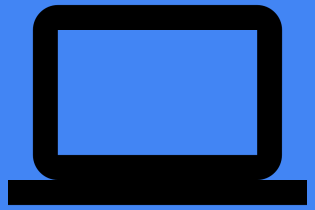
Tyrion



DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

Arya



Jon

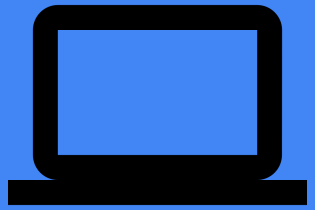
PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp
SRV	tyrion.local 8888

SRV
tyrion.local 8888

Tyrion

DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

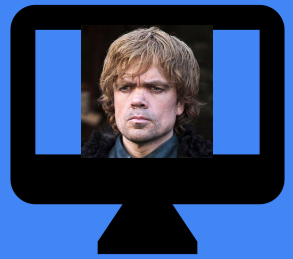


PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp
SRV	tyrion.local 8888

Arya

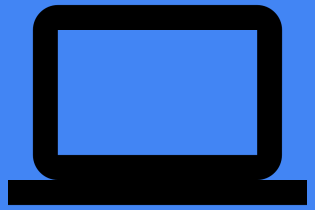
Jon

Tyrion



DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888



A?
tyrion.local

Arya

Jon

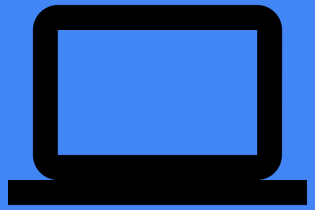
Tyrion

PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp
SRV	tyrion.local 8888

DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

Arya



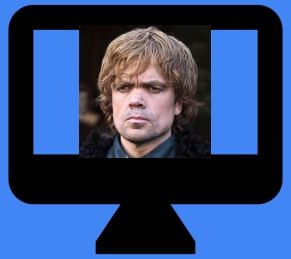
Jon

PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp
SRV	tyrion.local 8888
A	172.28.7.90

A

172.28.7.90

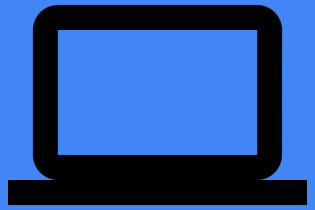
Tyrion



DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

Arya



Jon

A

172.28.7.90

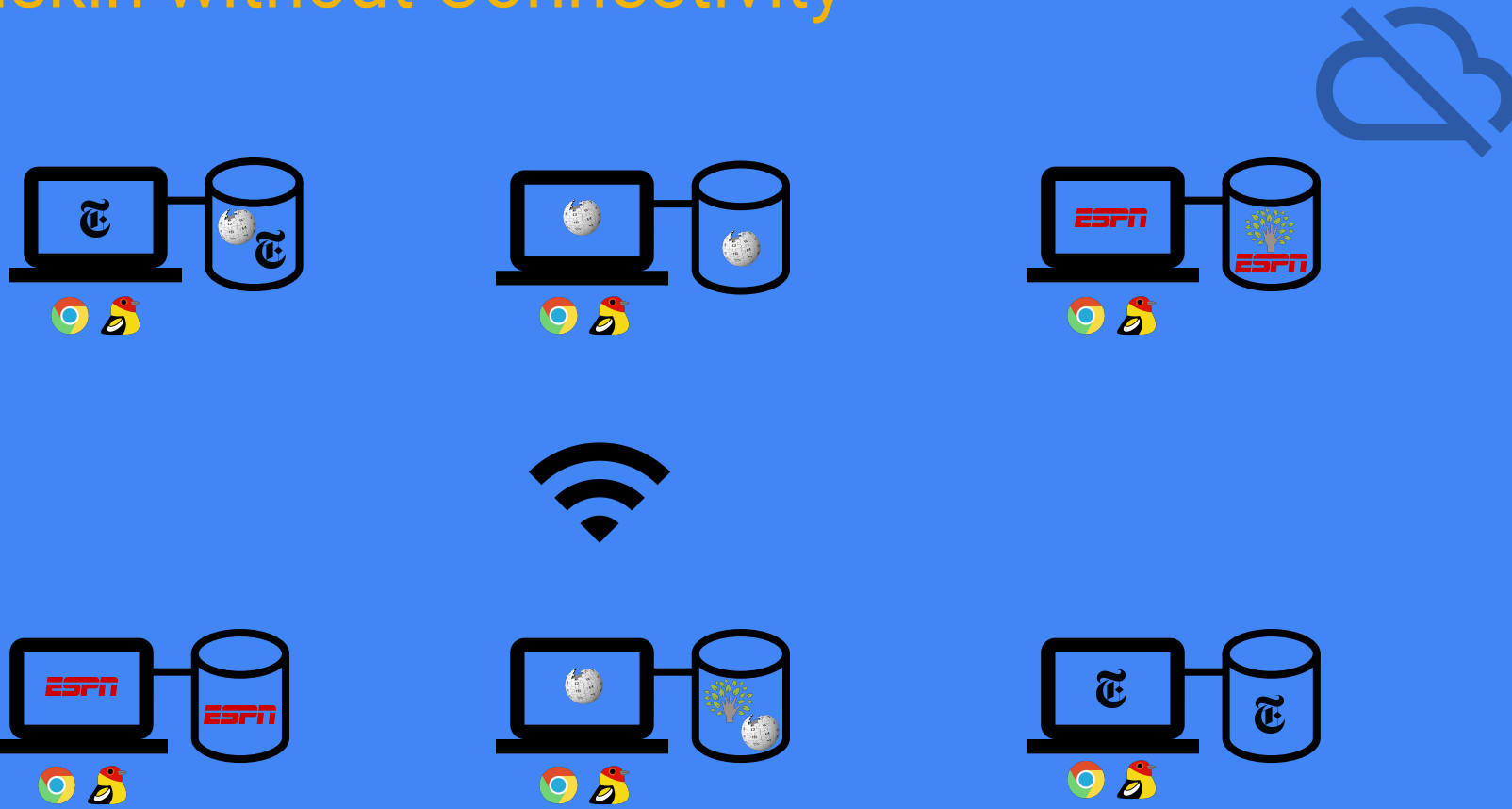
Tyrion

PTR	Arya.siskin.tcp
PTR	Tyrion.siskin.tcp
SRV	tyrion.local 8888
A	172.28.7.90

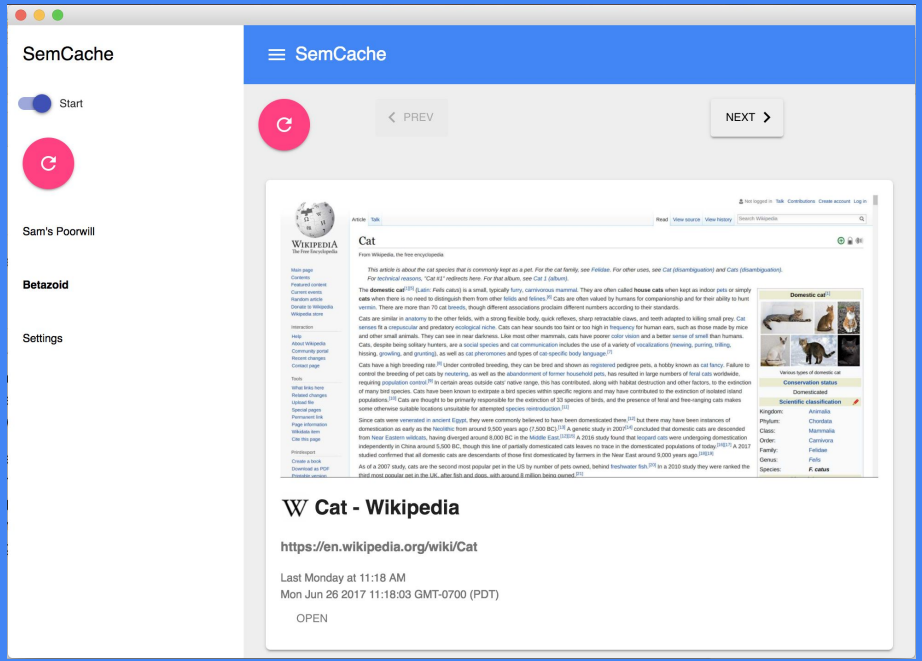
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer

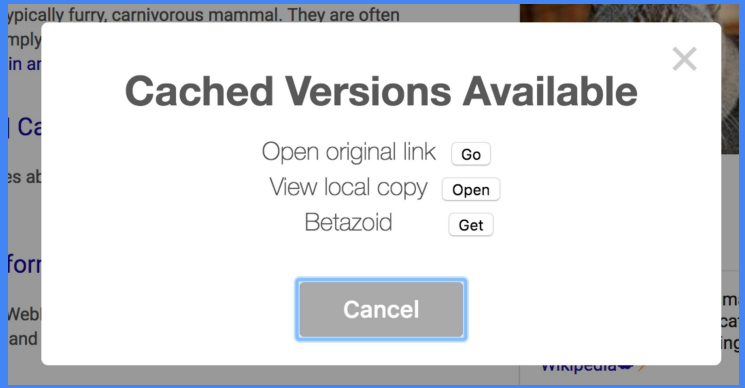
Siskin without Connectivity



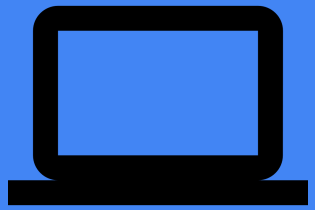
List Content



Annotate Links

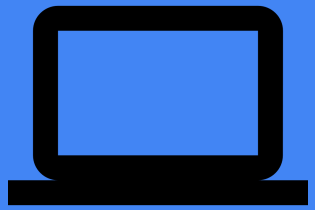


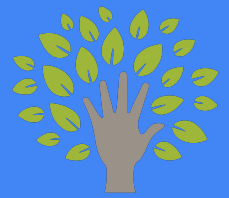
List Content

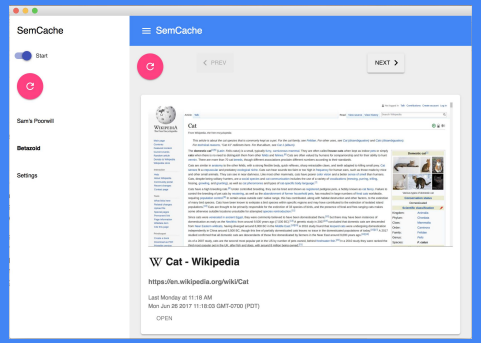


List Content

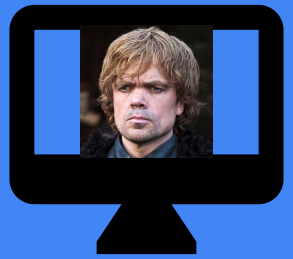
khanacademy.org/math1
nytimes.com/story1







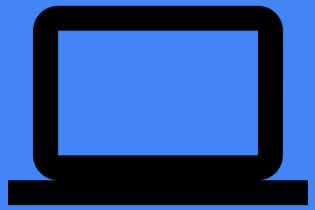
wikipedia.org/Cat

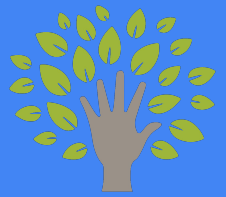


Annotate Links: naively

Cat - Wikipedia
https://en.wikipedia.org/wiki/Cat

khanacademy.org/math1
nytimes.com/story1





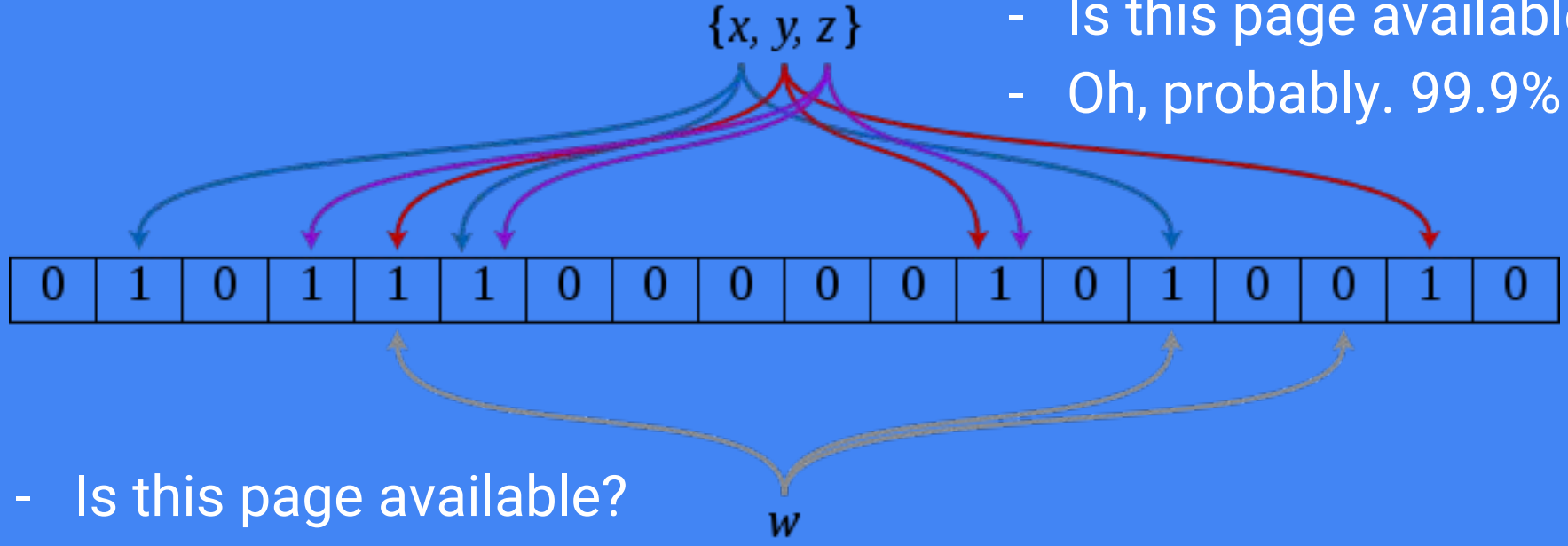
khanacademy.org/math1
nytimes.com/story1
wikipedia.org/Cat
…
…
…
…
…

wikipedia.org/Cat

Hard on bandwidth

1000 pages * 51 characters = 51 kB
10 peers * (51 * 9) = 4.59 MB
40 peers * (51 * 39) = 79.59 MB

Bloom Filter



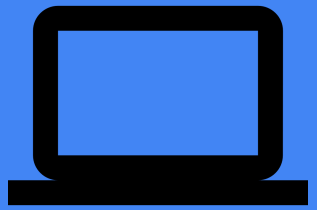
- Is this page available?
- Oh, probably. 99.9%

- Is this page available?
- Definitely not.

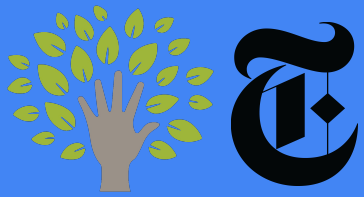
https://en.wikipedia.org/wiki/Bloom_filter

Bloom Filter

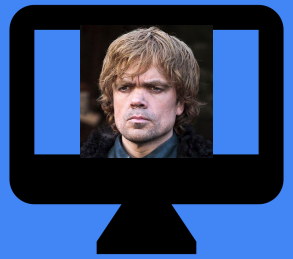
Cat - Wikipedia
https://en.wikipedia.org/wiki/Cat



0	1	1	0	0	1	0



1	0	0	1	1	0	0



0	1	1	0	0	1	0

1	0	0	1	1	0	0

Better on bandwidth

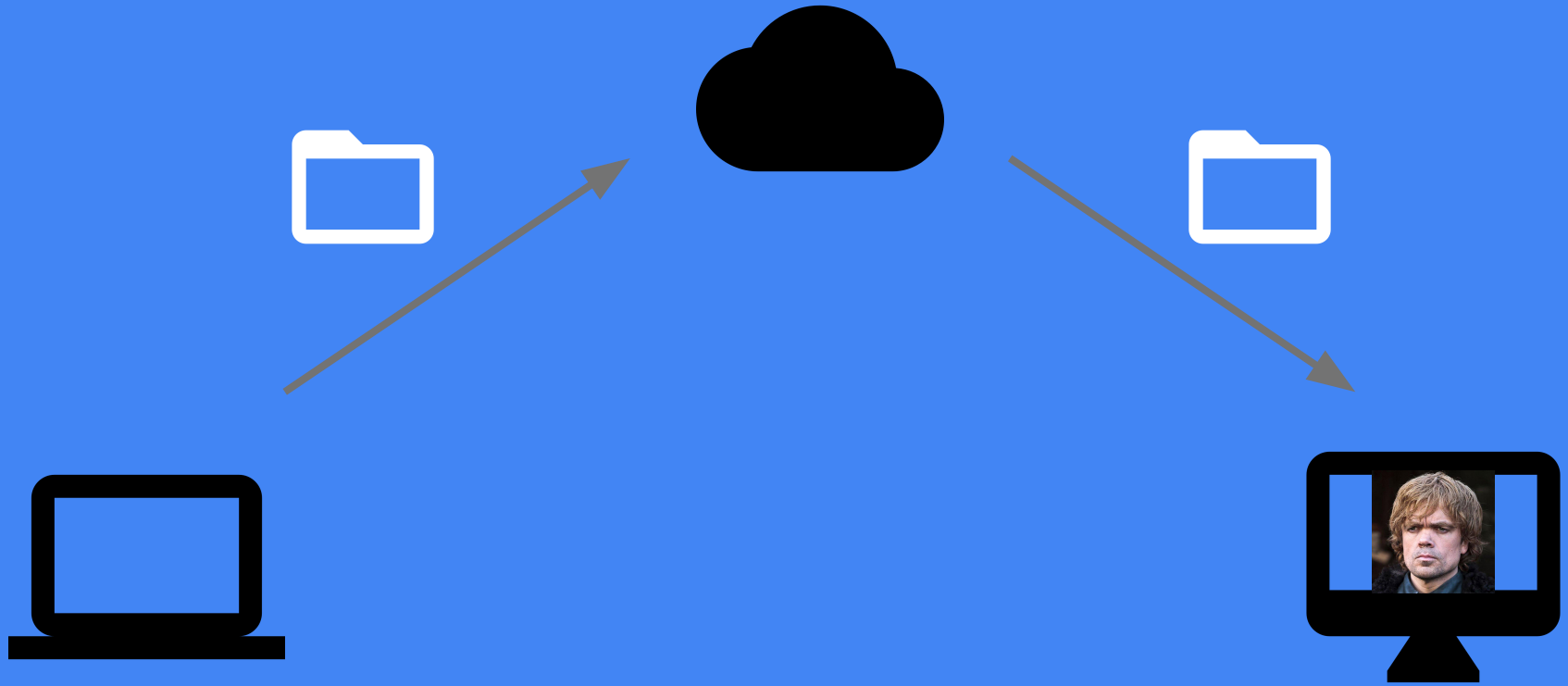
1,000 pages, 0.001 FP = 1.8 kB
10 peers * (1.8 * 9) = 162 kB
40 peers * (1.8 * 39) = 2.8 MB

80 MB to 3 MB with 40 peers

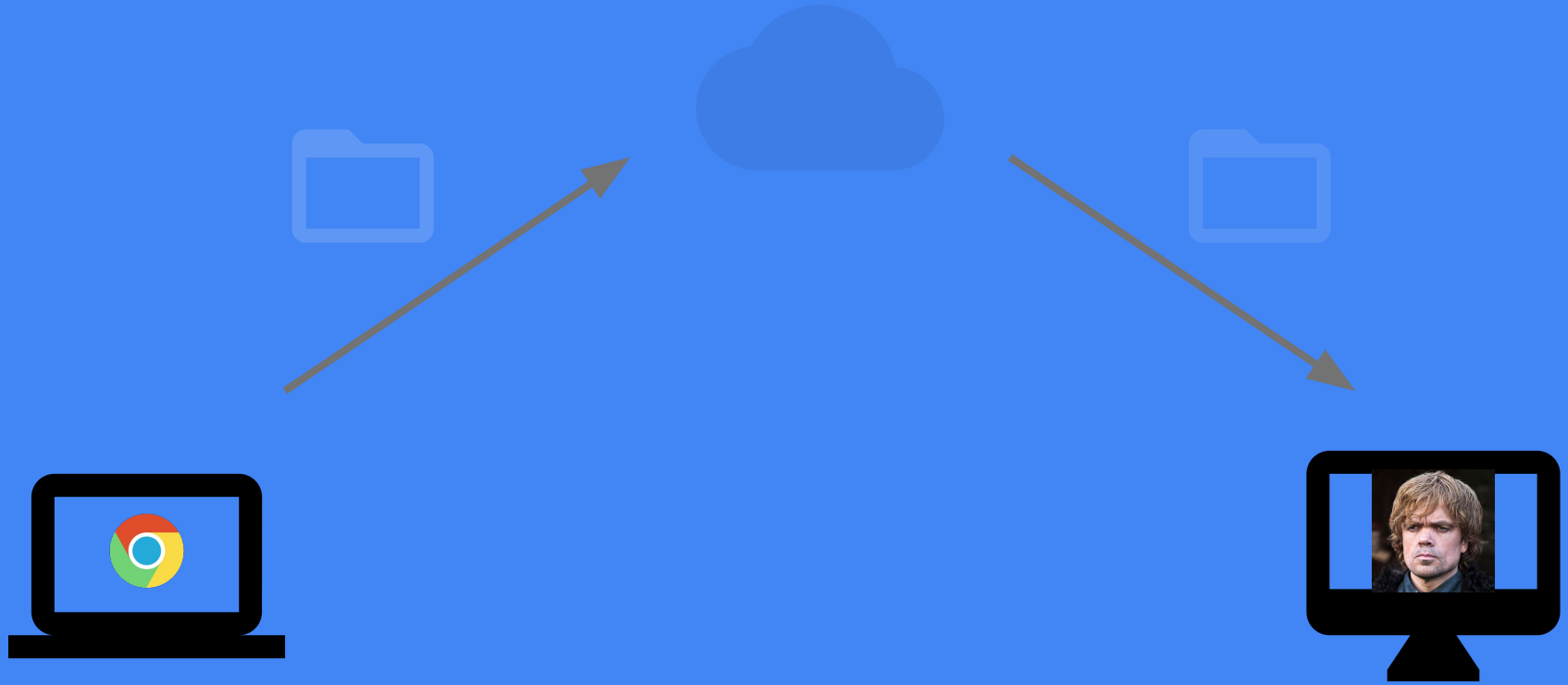
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer

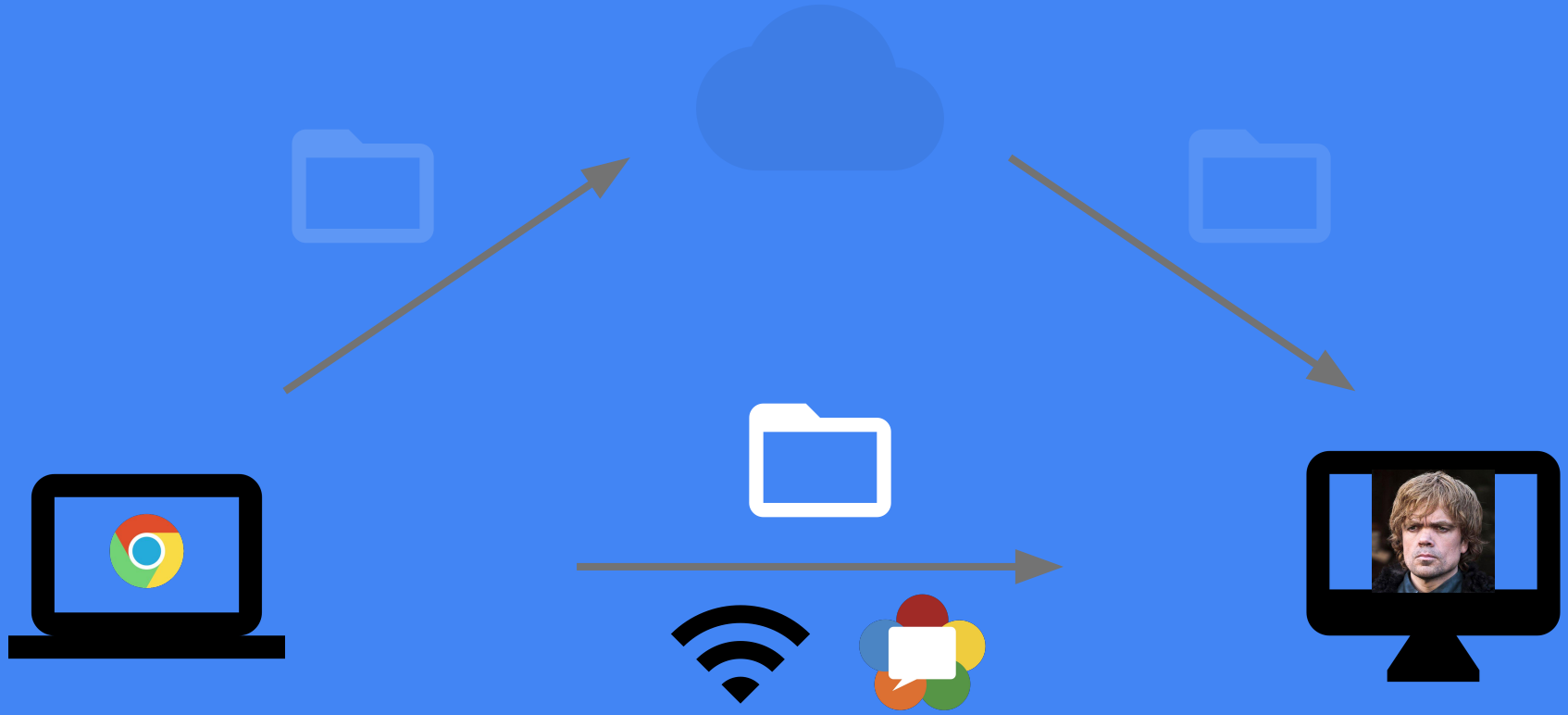
Conventional browser-based file sharing



WebRTC



WebRTC

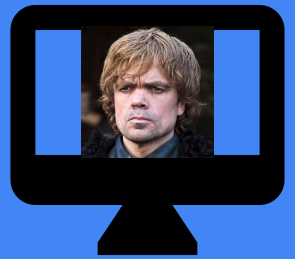


WebRTC

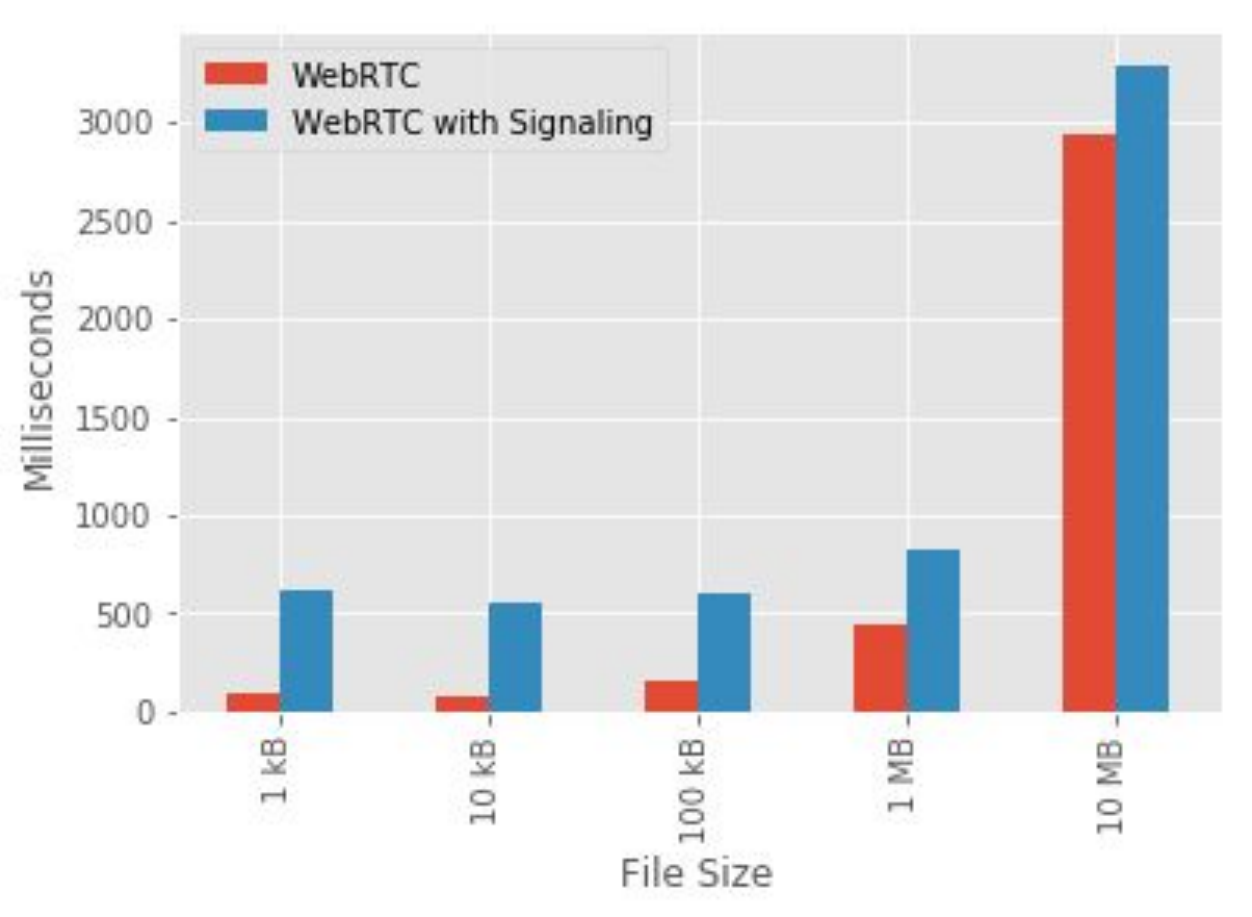
WebRTC?

172.28.7.90:8888

WebRTC.



Transfer Speed



What did we learn?

- Browser as infrastructure
  - One-click installation
  - Distributed cache
- Augment the natural browsing experience
  - Browse naturally, stay local when possible
- Static snapshots are imperfect
  - Place for improvement
  - Progressive Web Apps potential way forward

Thank you!
Questions?

github.com/srsudar/SemCache