Spotify infrastructure

Behind the scenes

Javier Ubillos

2012-05-16

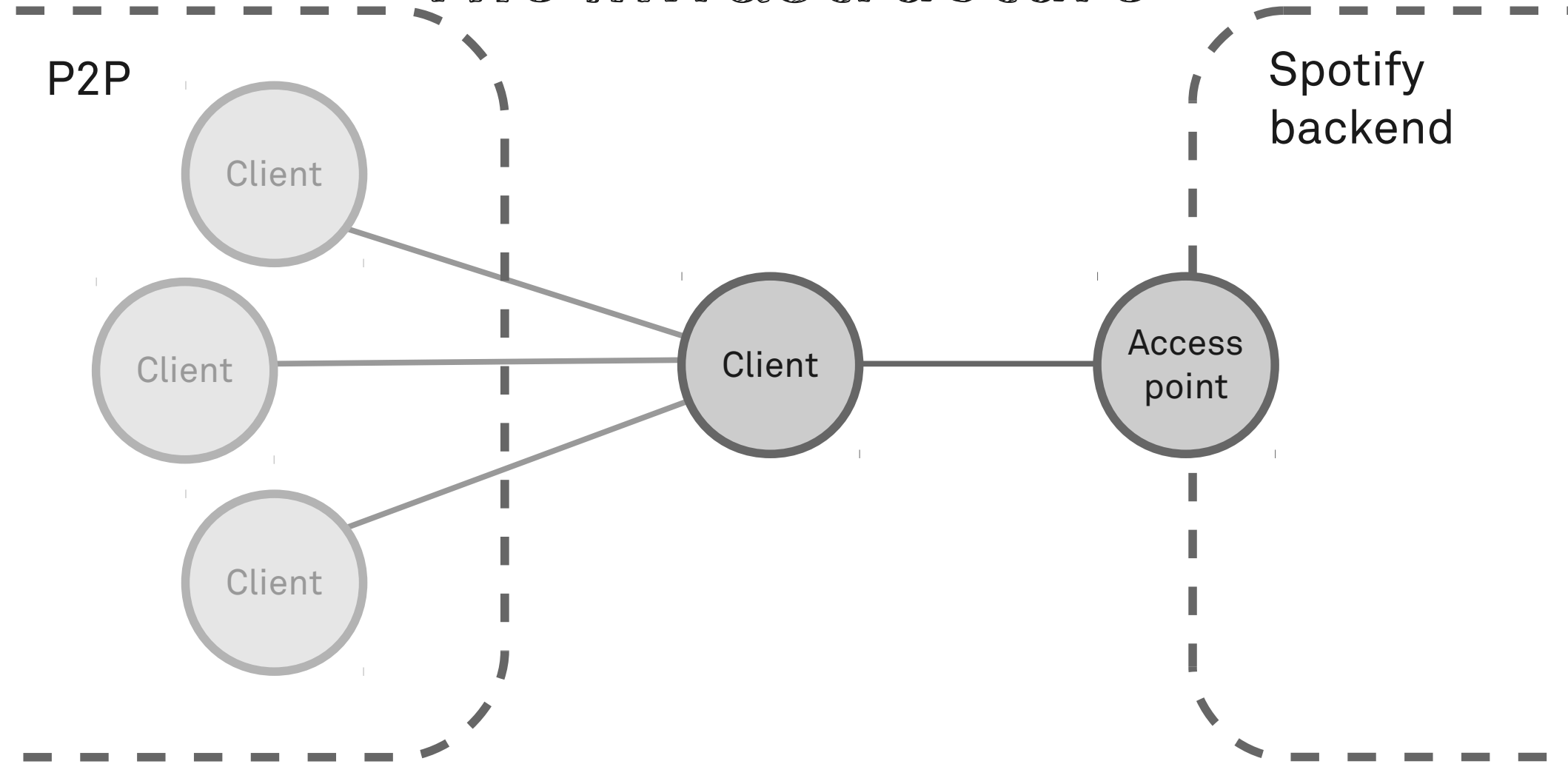
What is Spotify?

- Lightweight on-demand streaming
- Over 10 Million active users
- 18 Million tracks (Available in Europe)
- Available in 13 countries
  USA, UK, Sweden, Finland, Norway, Denmark, France, Spain, Belgium, Germany, Austria, Switzerland, The Netherlands and Germany.
- Convenient (more so than piracy)
- Legal

More numbers

- Over 20,000 new tracks added each day
- Over 700 million playlists created
- Around €170 million/$250m has now been paid to rights holders since launch in October 2008
- Spotify is the overall number two digital service in Europe, after iTunes. (IFPI Digital Music Report, Jan 2011)

The infrastructure



Latency is everything

- Google websearch: Adding 100ms – 400ms latency to serve the searchresults increases drops in searches from -0.2% to -0.6%

  - Speed Matters for Google Web Search, Jake Brutlag, 2009

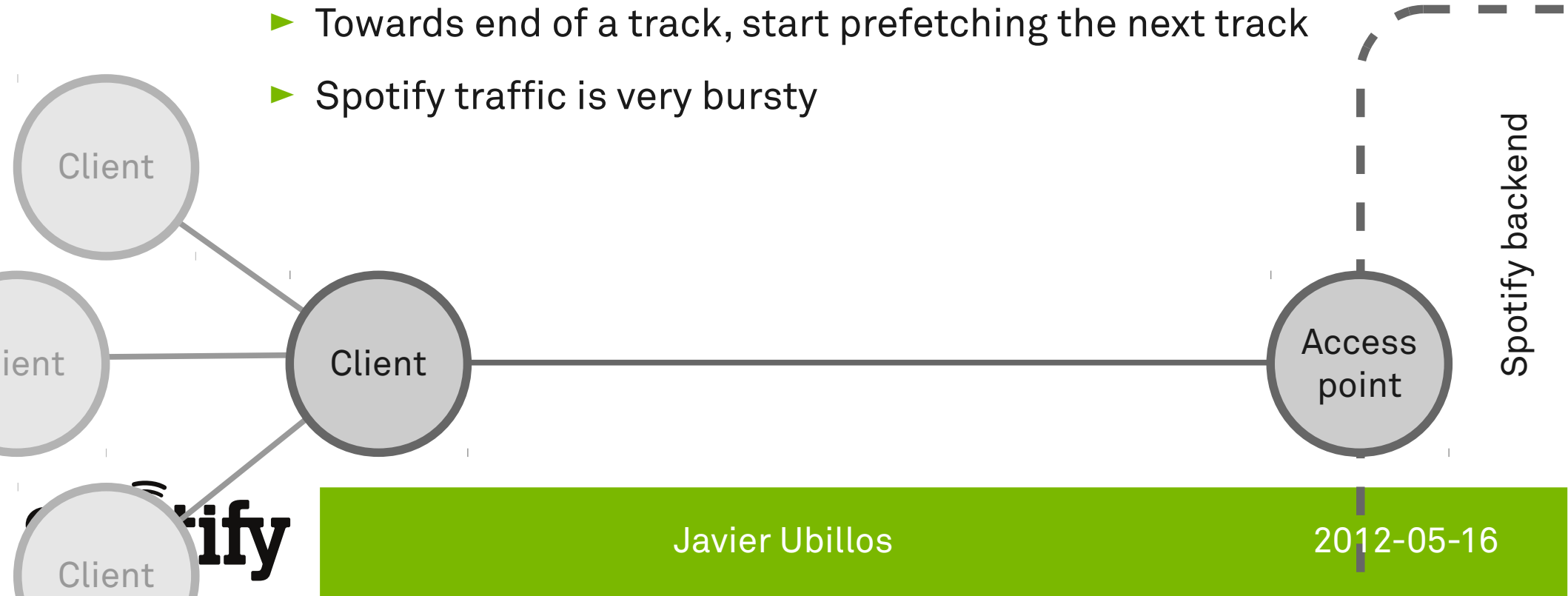
- Amazon reports that every 100ms of latency costs 1% of profit
- Google maps found that an increase from 400ms to 900ms to render search results dropped traffic with 20%
- A blink of an eye takes between 300ms and 400ms
- Median playback latency in Spotify is 265ms (feels immedate)

Caching

- Player caches tracks it has played
- Default policy is to use 10% of free space (capped at 10 GB)
- Caches are large (56% are over 5GB)
- Least Recently Used policy for cache eviction
- Over 50% of data comes from local cache
- Cached files are served in P2P overlay

Transfer strategy

- Fetch first piece from Spotify servers
- Meanwhile; search P2P for remainder
- Switch back and forth between Spotify servers and peers as needed
- Towards end of a track, start prefetching the next track
- Spotify traffic is very bursty

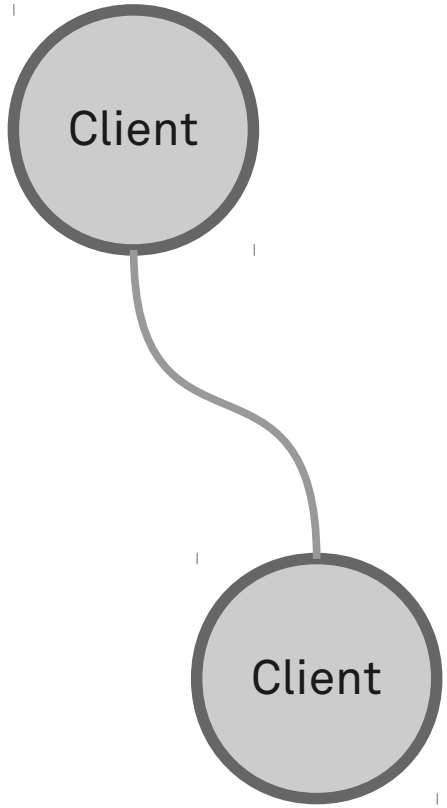


The parts

- The streaming protocol
- The P2P net
- The backend

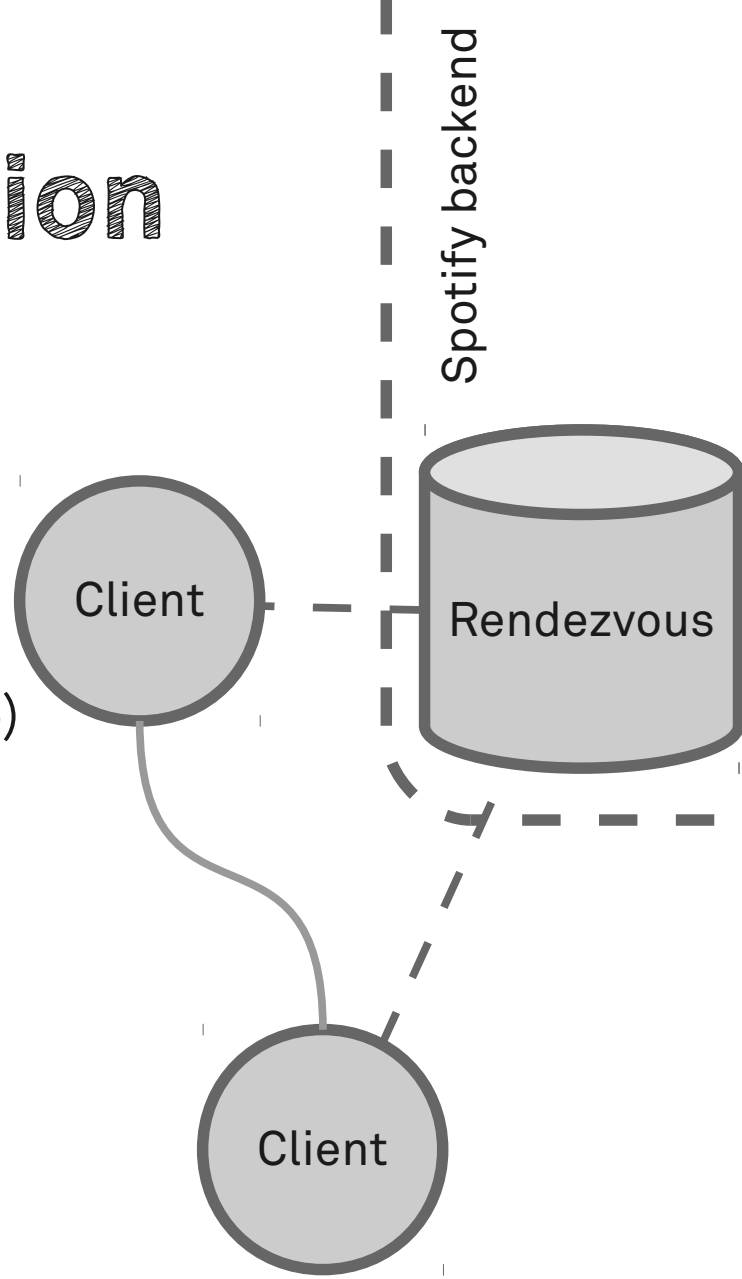
P2P overlay

- One P2P overlay for all tracks (actually three)
- Unstructured network (not DHT)
- Do not enforce fairness (e.g. tit-for-tat)
  - No pro-active caching
  - I.e only ever upload tracks you have listened to
- All peers are equal (no supernodes)
- Nodes who are not active (do not play) drop out of the overlay

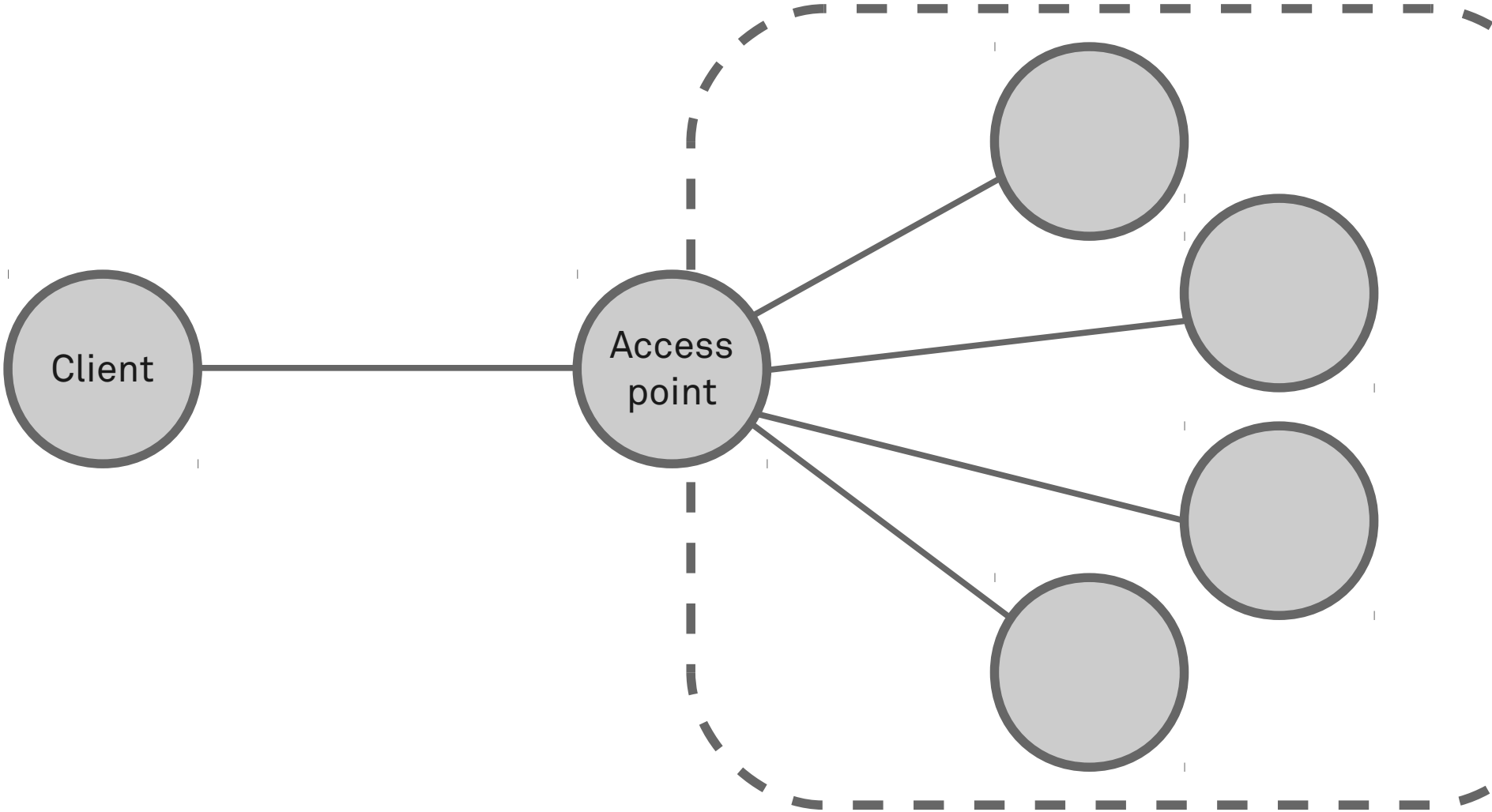


Peer coordination

- Server-side tracker (BitTorrent style)
- No overlay routing (e.g. Kademila)
- Broadcast query in small (2 hops) neighborhood in overlay (Gnutella style)
- Server-side tracker
- Gnutella style queries (2 hops)
- LAN peer discovery

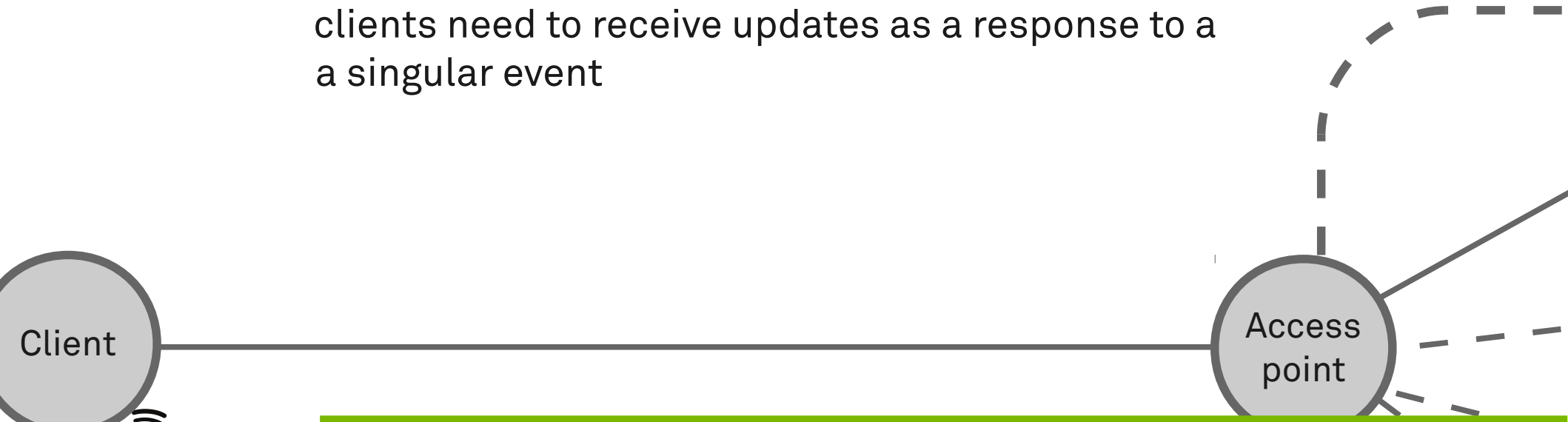


The backend



The backend

- Built on many small services
- All traffic enter and exit via the Access Points
- Try to keep service interdependence low
- (Lots of) Small services
- More and more of “one to many” traffic, where many clients need to receive updates as a response to a a singular event



Data collection

- Four general monitoring mechanisms
  - Plain logging of events
  - Probabilistic logging (mainly to avoid data transfer overhead in the client)
  - Munin (one datasample every fifth minute)
  - In house built statistics library

Data collection

- For services in the backend we use munin and RRDs*
  - Every datasource is sampled once every five minutes
  - Collected by site-local data-collectors
  - Forwarded to central data-repository
  - Graphed and presented to service owners and operations
- For client data and non-realtime data we store logs in a cluster
  - Not realtime, but does allow larger map-reduce jobs

* Round Robbin Database
Read/write data-store of constant size

Monitoring limitations

- Is a single sample, every fifth minute enough
  - Are the RRD resolutions good enough?
    300s, 1800s, 7200s, 86400s (one day)
    Five hundred samples in each interval
  - What to do with the data?
    - Currently, we use data and graphs for “after the incident” analysis and in helping us to locate the problem
    - Perhaps some predictive methods rather than after the fact analysis

Social monitoring 2.0
(+ arbitrarily chosen buzzwords)

- We have no idea how to proceed
- Let the users decide!
- Interactive front-end
  - Interactive graphs and dashboards
  - Shareable and collaborative

Thank you!

Questions?

jav@spotify.com

Data sources

- 55.4% from local caches
- 35.8% from P2P
- 8.8% from Spotify servers

- Median latency 265 ms
- 75th percentile: 515 ms
- 90th percentile: 1047 ms
- Less than 1% of playbacks had stutter occurrences.