Why WebRTC is the way it is

Some History
from someone who was in the room.

Tim Panton CTO |pipe| - tim@pi.pe

WHY?

The number one question asked
by webdevs when they first encounter
WebRTC

WTF?

The second question asked
by webdevs when they encounter WebRTC

My goal is to Explain the Inexplicable

Why did it take so long?
libwebRTC was open sourced in 2011

- Optimisim
  - We have everything we need
    (Google had bought a tonne of relevant IPR)
  - We have been doing this for year
    (Cisco had lots of SIP products)
  - All the RFCs already exist
    (Reuse existing protocols)
  - Done in 18 months.

Why did it take so long?

libwebRTC was open sourced in 2011

- Ignorance
  - It’s just a phone in the browser
  - Like 800 numbers
  - Or skype perhaps
  - It’s an RTC problem

Wrong!

Google should have bought phonefromhere.com :-)
We knew.

Who ?

- Google
- Cisco
- Ericsson
- Microsoft
- Mozilla
- Voxeo (me as spear carrier)

W3C and IETF

Why no standardised signalling?

- SIP vs XMPP vs H323 - no easy winner
- Auth — web auth isn’t like SIP auth
  - Need to tie web sessions to temp SIP ids
  - Grief to manage
- Orchestration
  - Easier to tie call state to web app state
    if web app does call control

Why P2P?

- Skype was the dominant ‘competition’
  - At that point in time it was a P2P protocol
  - On a mesh framework
- Majority of those involved were SIP folks
  - SIP had always aspired to be P2P
  - It just never worked that way
- So it seemed natural back then.

Why E2E (DTLS/SRTP) ?

- Snowden.
  - Several IETF folks took this personally
  - Proved to be convenient interception points
- Rearguard action to mandate SRTP/SDES
  - ‘Too difficult - can’t be done’ etc
  - 4 of us implemented DTLS/SRTP
in Java in a few weeks

Why RTP ?

- Standards politics.
  - Adobe too slow
  - Skype owned by MS
  - IAX2 only an informational RFC

Why Datachannels ?

- Our experience was that in call data was useful
- No DTMF isn’t enough!
- RTP datachannels super clunky
  (Still supported in chrome
  8 years after being deprecated)
- SCTP suitable (if overkill) with an RFC

Why so many optional modes ?

- Early media, bundle, SDES, PRANSWER etc
- ‘Future Legacy Illusion’
- The idea that there were existing telco middle boxes that would support webRTC unchanged
- P2P and E2E ensured that could never be true. But we still have the APIs.
- Complex to learn and a testing nightmare

Why offer/answer semantics ?

- Because they didn't know better ?
- 3rd party client illusion
  - Facebook <->Myspace calls
  - Like the phone system
- CORS and Biz realities
  mean that both sides share
  a state machine in the web service.

Why those codecs ?

- Silk’s success -> opus
- License counting is harmful -> VP8
- But hardware -> H264
- Future legacy interop -> ulaw,g722 etc

Perhaps the only acceptable bit of optionality IMHO

Why Bother?

Huge success

- Billions of users
- Billions of minutes
- On (almost) every smartphone
- W3C REC last week !
- IETF cluster 238 !

Thanks for all the hard work!

Demo
(I kinda have to)

- H264 from raspberry pi
- Controlled over data channel
- Audio conference
- In a browser
- Letting kids play/interact in lockdown
- Not in the original spec ;-)

tim@pi.pe @steely_glint on twitter