draft-ietf-stir-oob-01
Out of Band

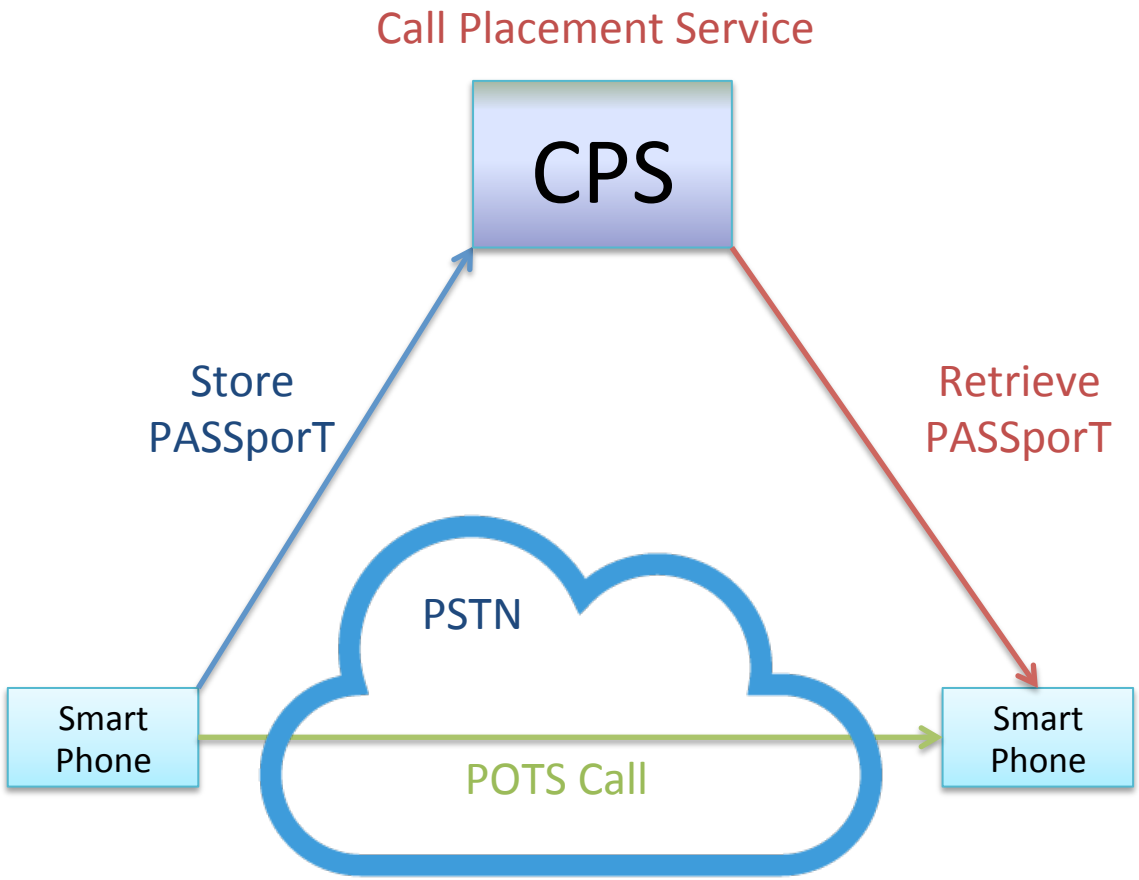
STIR WG IETF **100** Singapore

Nov 2017

Limits of RFC4474bis

- It's in-band – end-to-end IP-IP
  - At best, it addresses the SIP-to-SIP use case
  - Not going to help with SIP-to-PSTN, PSTN-to-PSTN
    - Import for transitional adoption, legacy networks, enterprises, etc.
  - We did in-band first because existing deployments need it
    - Like the IPNNI, now the SHAKEN profile
- Even some IP-IP deployments may not pass Identity e2e
  - Difficult to anticipate what will survive administrative boundaries
    - You can understand "boundaries" pretty broadly
  - And some existing deployments might just block Identity
    - As they block all new headers; especially B2BUAs

Basic STIR Out of Band

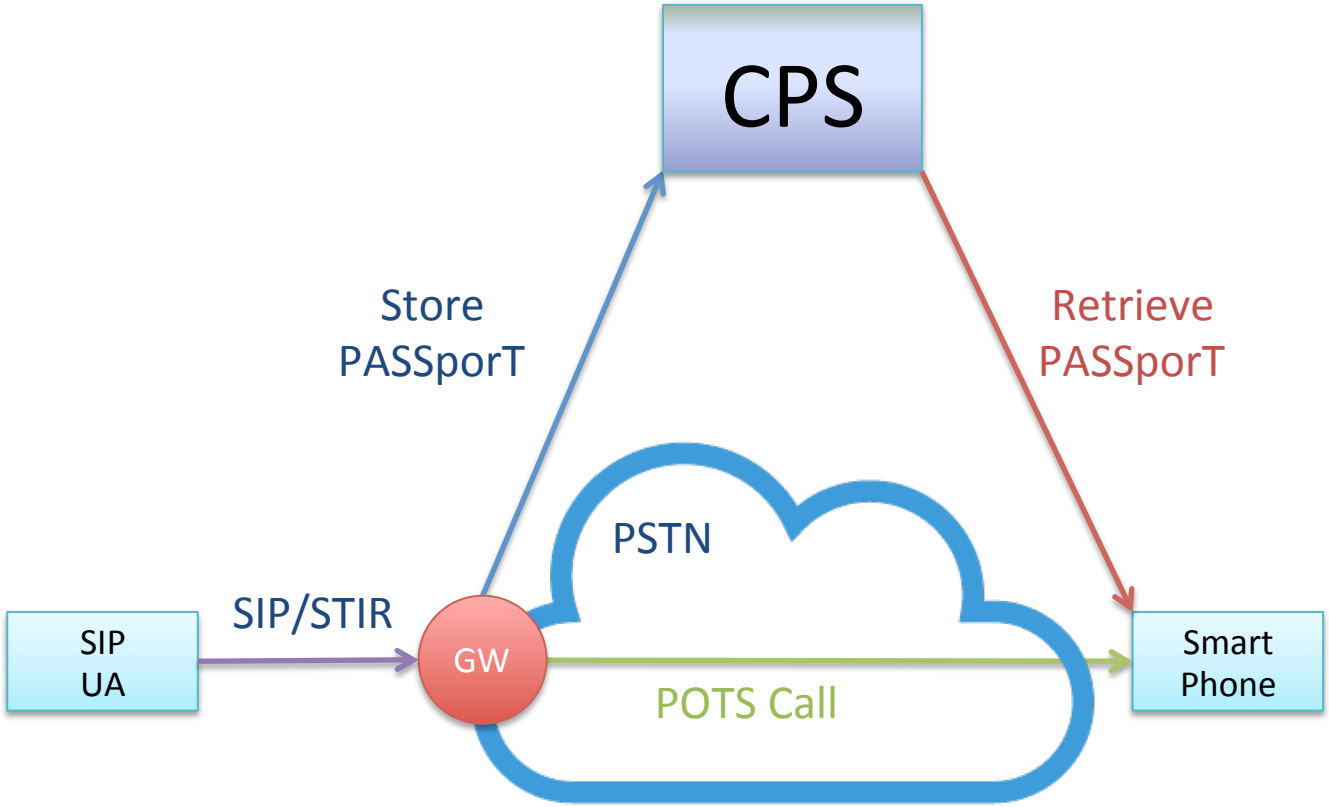


Smart Phones are not just mobile phones, and not just end-user devices

Obvious Questions

- Okay, how does the originating side know where to find a CPS?
  - And how do we make sure the terminating side comes to exactly the same conclusion?
    - Need a CPS discovery mechanism
    - A few initial ideas in the draft now – not the focus today
- How do we make sure the right parties store and retrieve PASSporTs from a CPS?
  - Mostly, to manage the risk that someone other than the called party will fetch them? Or just record who fetched what?
    - Significant privacy concerns
- These are the things we're trying to lock down now

Who can put and get PASSporTs?



This is why it's hard to require authorization for storage

Also Tricky: Optimizing for Privacy

- We want to minimize potential metadata collection
  - Give the CPS as little insight into calls as possible
- The called and calling party should have no required preassociation
  - Except as needed for key discovery and CPS discovery
    - We are assuming both sides have STIR credentials
- Need some way to store PASSporTs such that they can be found
  - CPS needs to index store PASSporTs based on some public fact about the called/calling parties
  - e.g. “Give me PASSporTs for the called number (me?)”

Overview of the Approach

- Allow anyone to store encrypted PASSporTs, indexed at the CPS by the called party's public key
  - PASSporTs are encrypted with a key of the target
    - CPS cooperates with a cert cache, allows retrieving of public keys by target TN
      - Might give you multiple keys for the same TN: carrier, reseller, user, etc.
- Allow anyone to retrieve any PASSporTs
  - CPS always returns at least one encrypted blob when asked for a PASSporT for a given public key
    - Whether there is a call in progress or not
  - Only the intended recipient will be able to decrypt real PASSporTs and determine that there is a legit call in progress

Benefits of the solution

- Encryption really limits what the CPS sees
- Difficult to poll the server to learn about calls in progress
- Indexing by the public key, rather than the called party number, works better with multiple certs
  - Calling party may need to store multiple PASSporTs if multiple entities hold credentials for a number
    - Carrier, service bureau, enterprise, etc.

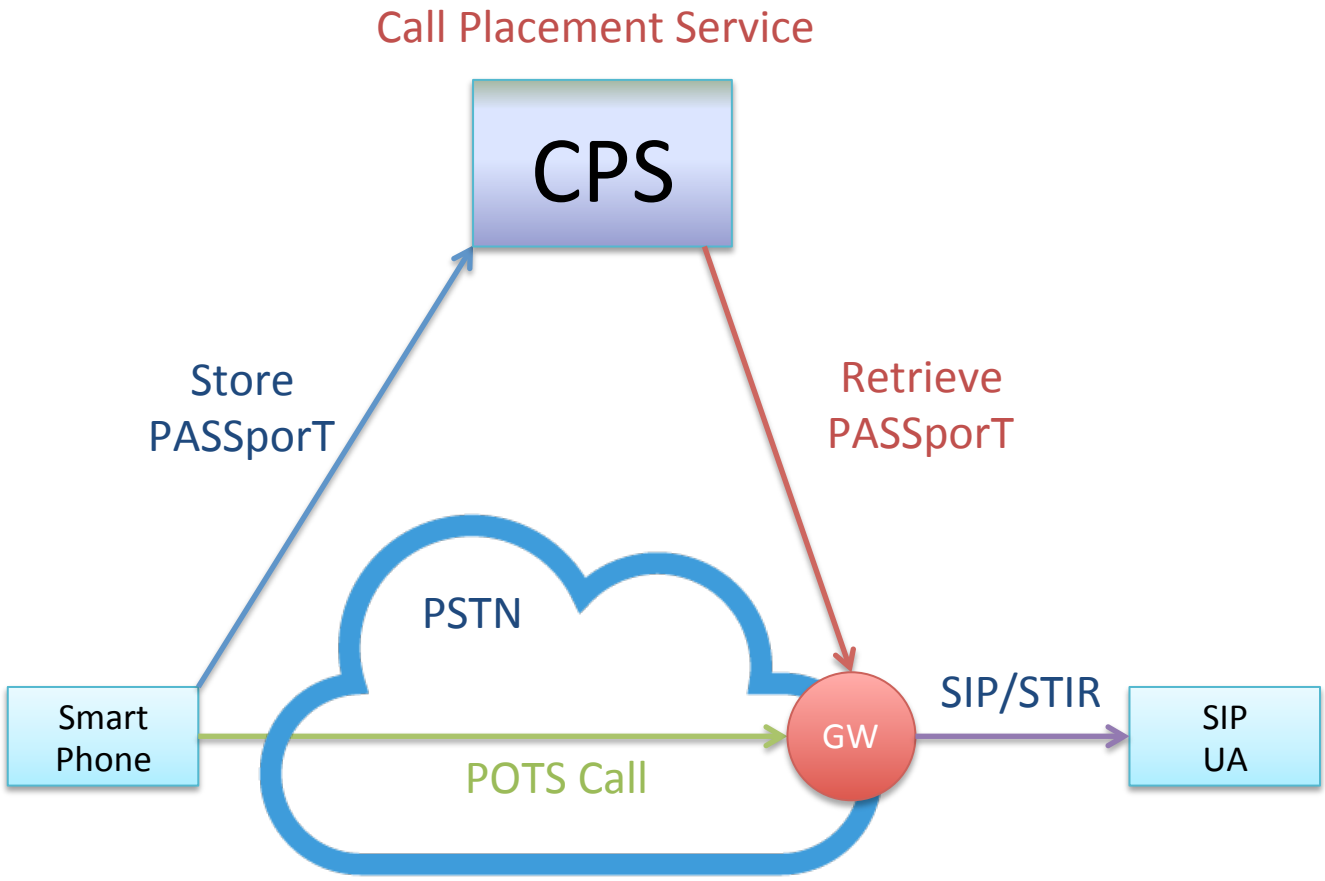
Flood prevention

- PASSporTs are signed, so it almost doesn't matter who stores them
  - Almost – need some kind of DDoS protection from attackers storing millions of bogus PASSporTs
- The authority to store might still require a STIR credential
  - But don't want to have to authenticate a storing party with a STIR cert, that reveals the calling party to the CPS
  - Possible to limit storage with some kind of fancy tokens based on having a valid STIR cert
    - Effectively pre-associate with the CPS before storing
    - Acquire a token you spend to store a PASSporT later
  - Ways to get this to work for gateways, even

Service Discovery

- How many CPSs should there be, and how to you find them?
- The more we “federate” the CPS function, the more pressing service discovery becomes
  - Less monolithic CPS means no single point of monitoring
    - But how can the caller and callee agree on which CPS serves both?
  - How much pre-association does a caller need to have with a CPS to place a call?
- Similarly need to discover a credential service for OOB
  - A STIR cert could contain a field for the CPS that services the called party
  - That would make this the familiar credential discovery problem
    - Not a solved problem for operations, but lots of candidate protocols

Also, what about this case?



Maybe a SIP Identity-Encrypted header? RCD might need it anyway

Next Steps

- Sound good?
- To Do – beyond just architecture
  - Need to specify at least one CPS discovery mechanism
  - Need to describe the storage/retrieval protocol
    - Pro tip: it's probably HTTP HTTP
  - Need to specify an OOB authentication and verification service procedure
    - Varies from RFC4474bis because that text is based on comparison to SIP fields
    - This needs to refer more abstractly to calls in progress and how the AS/VS does reference integrity