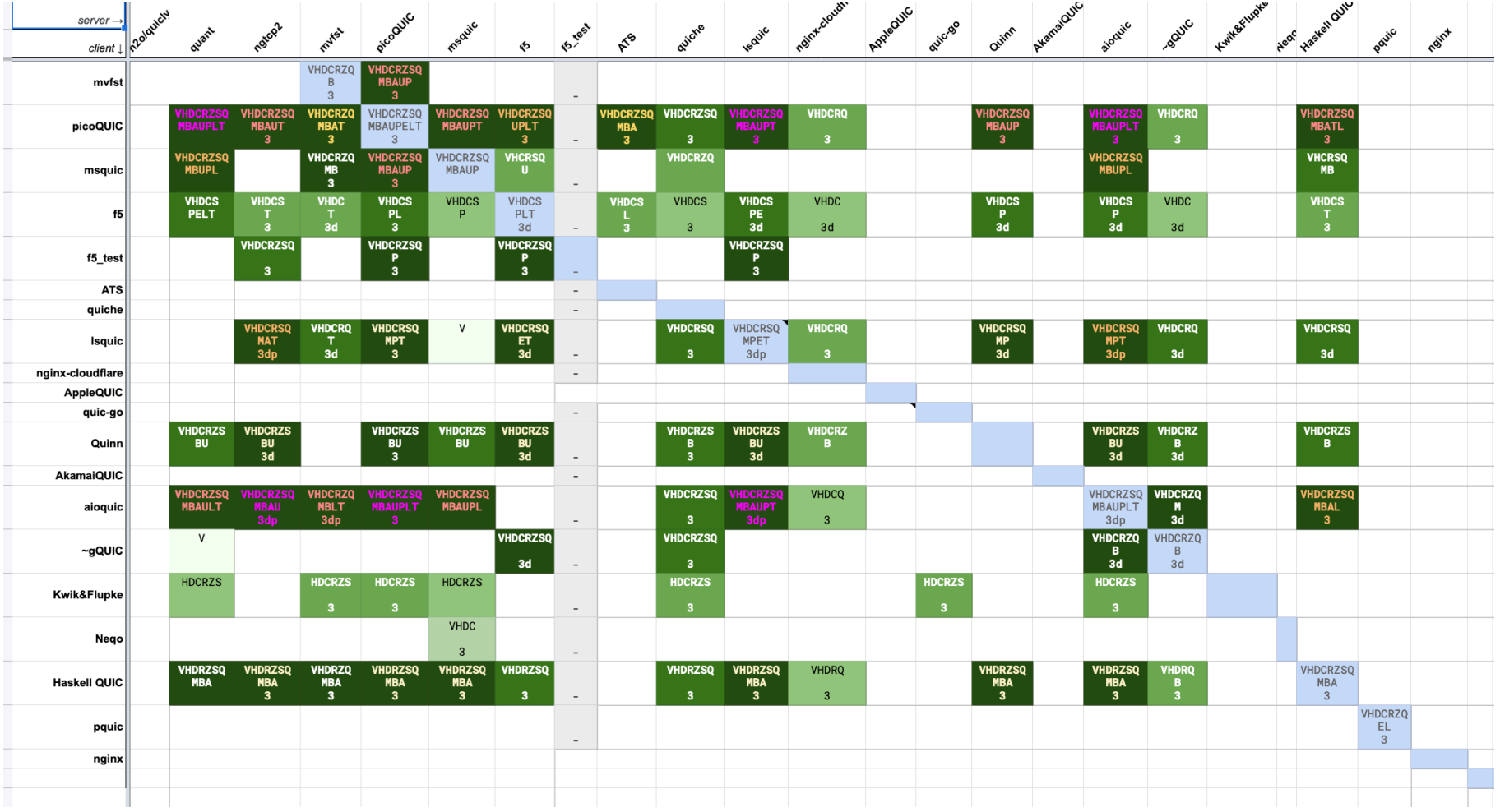
DMA Stakeholder Workshop: Interoperability

Eric Rescorla
CTO, Firefox and Internet Platform
Mozilla

A comparison point: interoperability for QUIC



Big Picture

- End-to-end encryption means end-to-end protocols/API
- Sender and receiving endpoints need to share
  - Key establishment protocols
  - Message formats
  - Voice/video protocols
- Some gatewaying is still possible
  - Identity/authentication system
  - Message transport
- These are very semantically complicated interfaces

Key Establishment

- Need some kind of key establishment protocol
  - Lots of cryptographic prior art here (OTR, Signal, MLS, ...)
  - Also lots of failures! Hard to design and implement
- This needs to be implemented in the **clients**
- What about group messaging?
  - Full mesh
    - Robust in “mixed” settings
    - Inefficient for large groups
  - Single group key
    - Requires a single protocol

Message and media formats

- This media is encrypted
  - So it can’t be translated
  - The sender needs to send exactly what the receiver can receive
- Lots of prior art here too (MIME, RTP, etc.)
- These pieces are also very technically complicated
  - Not as brittle as the cryptography
  - But very hard to get a high quality experience, especially for voice/video
- Need to define the core set of interoperable features and extensibility model

Identity

- Currently each gatekeeper is a closed identity system
  - Even if the names are externally meaningful (E.164 numbers)
  - Sometimes they overlap!
- How does system A authenticate its users to system B?
  - And how do they appear?
  - What do we do about overlapping identities?
  - Do you need to be able to detect which system an identity is on?
- Can we detect misbehavior?

Multiple gatekeeper scenarios

- Gatekeeper provided interfaces let new entrants interoperate with gatekeepers
  - By implementing those interfaces
- But what if there is more than one gatekeeper in a group?
  - Which interfaces/protocols does the group use?
  - Are gatekeepers required to talk to each other?
- Burden is likely to be on smaller players to make this work unless we have the right incentives

Suggested framework for interoperability (nonexhaustive)

- **Transparency**: Complete API/protocol specifications
- **Reliability:** Interfaces that remain stable when published with any major chances notified in advance
- **Testability**: Publicly available test servers (inc. log access)
- **Support**: Sufficient investment of resources (human and otherwise) to aid community with implementation
- … and others