Google™ 10 I/O

# Open Source Google Wave: Building Your Own Wave Provider

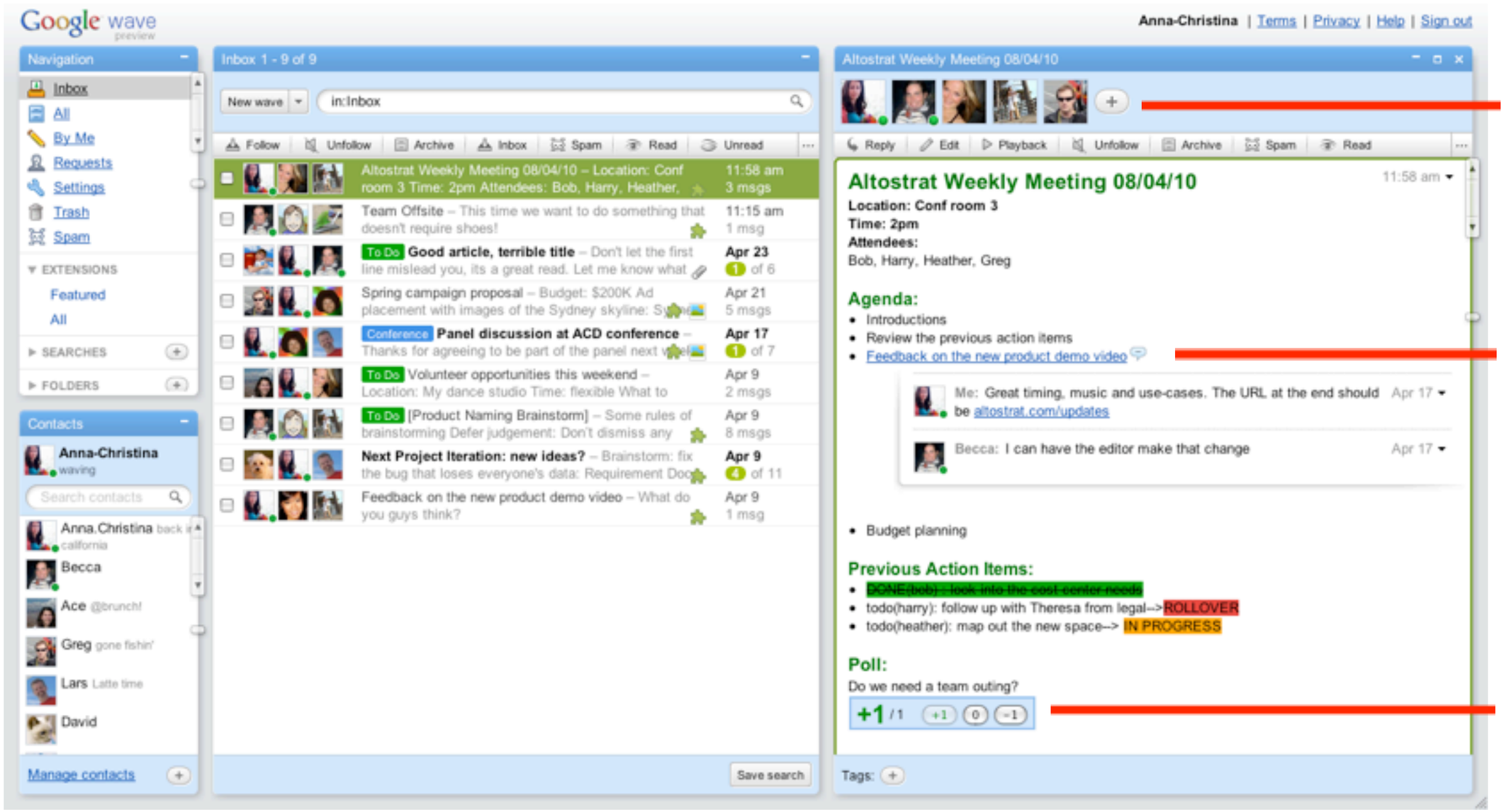
Dan Peterson
Jochen Bekmann
J.D. Zamfirescu-Pereira
May 19, 2010

# Agenda

- Introduction & Background
- Architectural Overview
- Open Sourced Code
- Federation Protocol
- Client/Server Protocol
- Wrap-up
- Q&A

- Wave: http://bit.ly/9s8fLO
- #Wave3

# Google Wave - Get stuff done with groups of people




Simple Sharing Model

Reply Anywhere

Live Editing

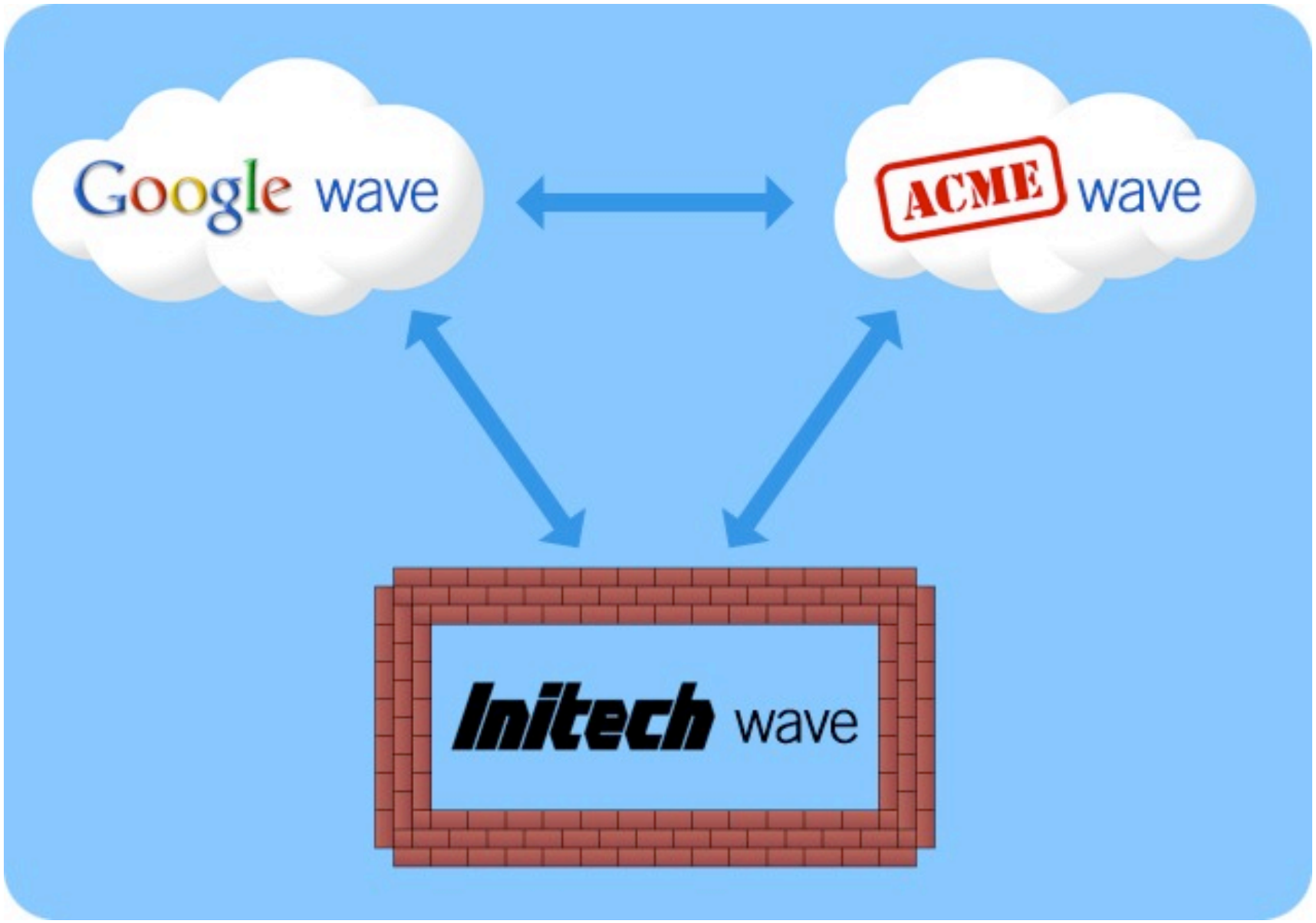
Custom Extensions

A wave is equal parts conversation and document

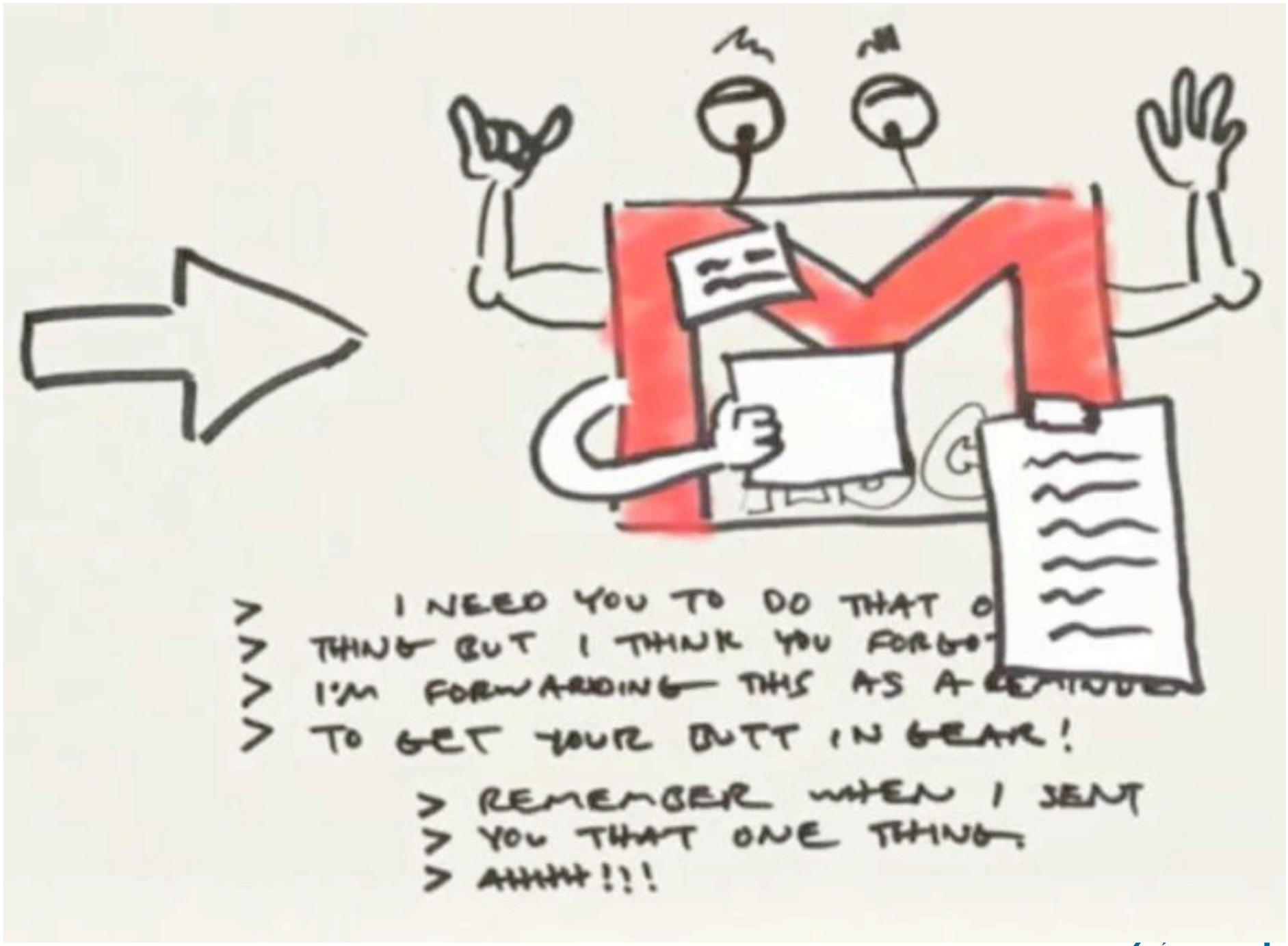
Google 10 I/O

# Google Wave Federation Protocol

Google I/O 10


Google wave
ACME wave
Initech wave

# How collaboration works today




Thanks to epipheo for the graphic!

# Wave Federation

- Enables user choice among wave providers
- Provides complete (data) control
- Iterating as an open specification

# Where we are today

- Published (draft) protocols & whitepapers
- Wave OT + FedOne + Document Model
  - 60K LOC, Java, Apache 2
- Opened wavesandbox.com federation port

# Today: Going further

- Published (draft) protocols & whitepapers
- Wave OT + FedOne + Document Model
  - 60K LOC, Java, Apache 2
- Opened wavesandbox.com federation port
- Concurrency Control + Wave Model
- Rich Text Editor from Wave Client
- Client/Server protocol for FedOne

# Wave Providers

- Novell Pulse
- QWave
- PyGoWave
- Ruby on Sails
- ARWave

# Today: More Wave Providers

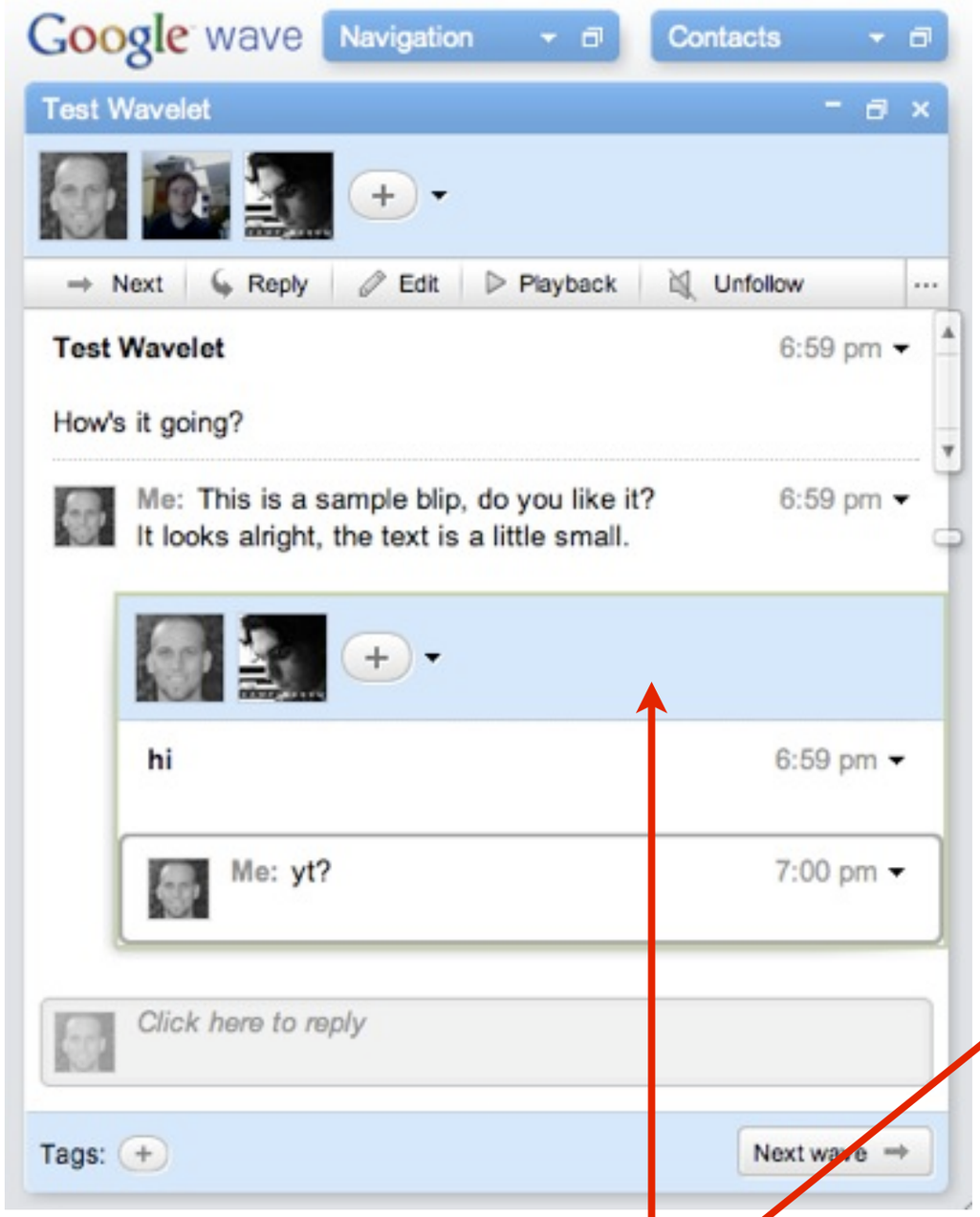
- Novell Pulse
- QWave
- PyGoWave
- Ruby on Sails
- ARWave
- SAP StreamWork
- ProcessOne OneWave

# Where we're headed

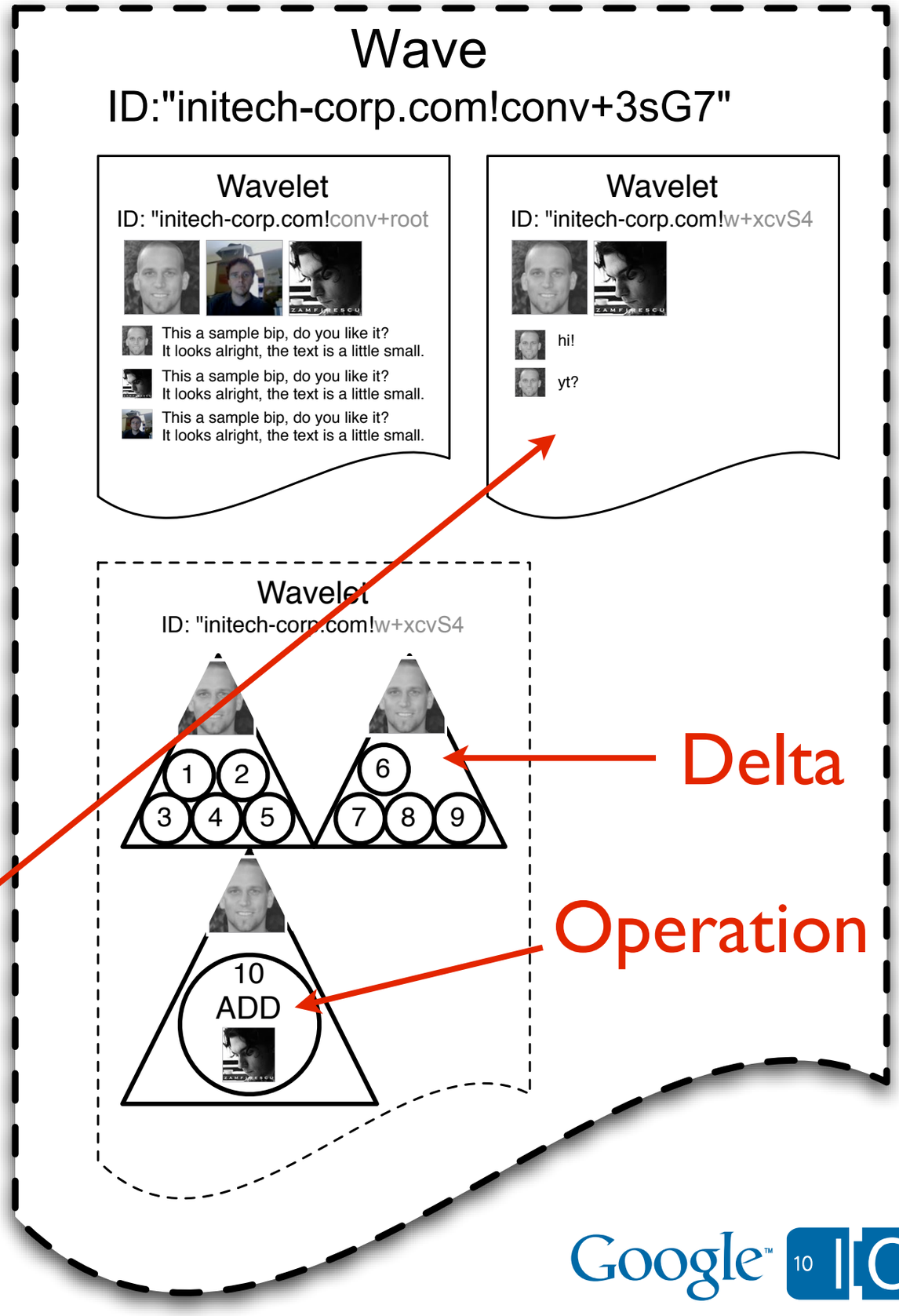
- Iterate and improve on the protocol specs
- Ship wave.google.com's federation port
- Build a production quality reference implementation
  - We're open sourcing more code; let's work together

# A Brief Architectural Overview

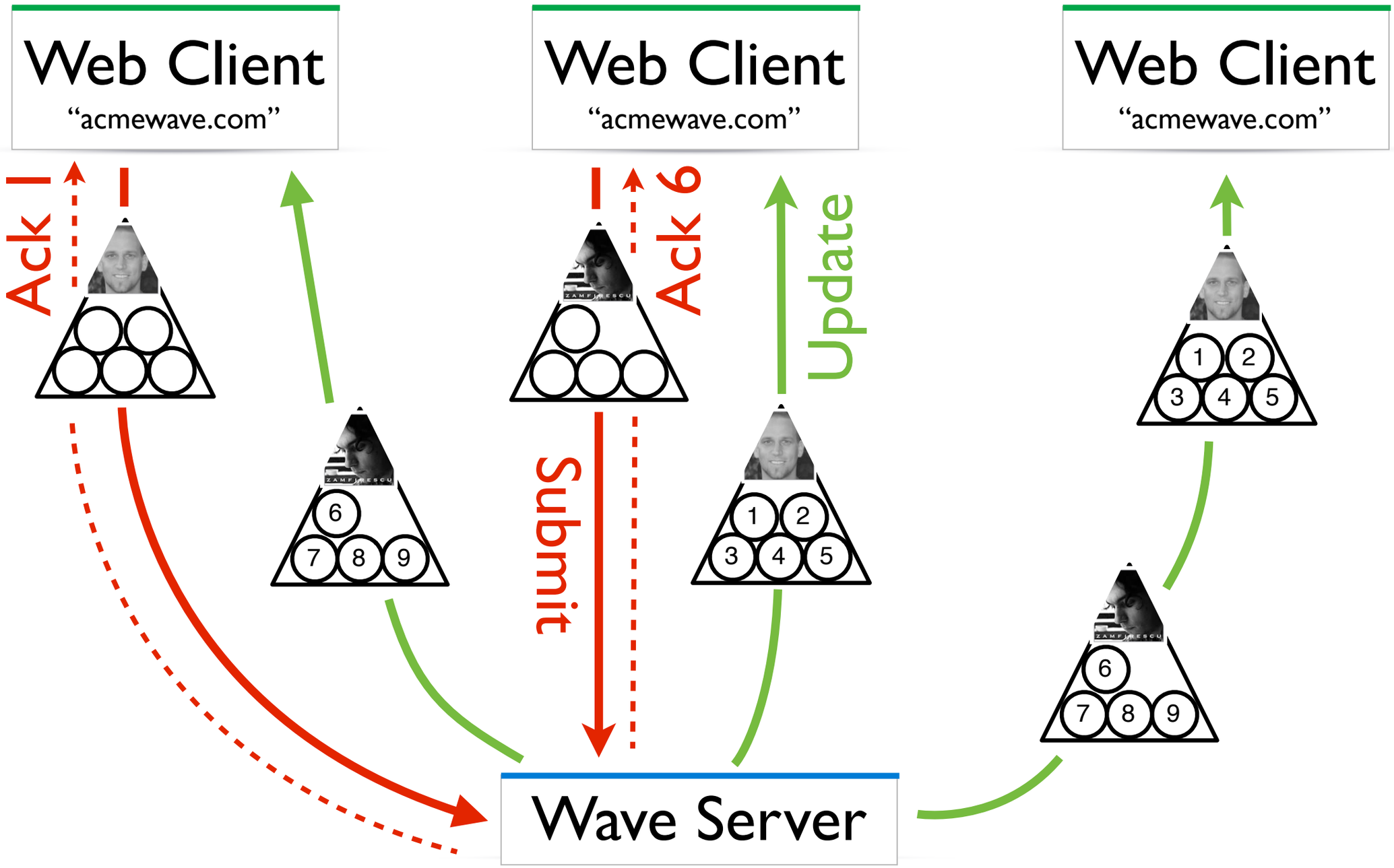
# Key Concepts




Private Reply

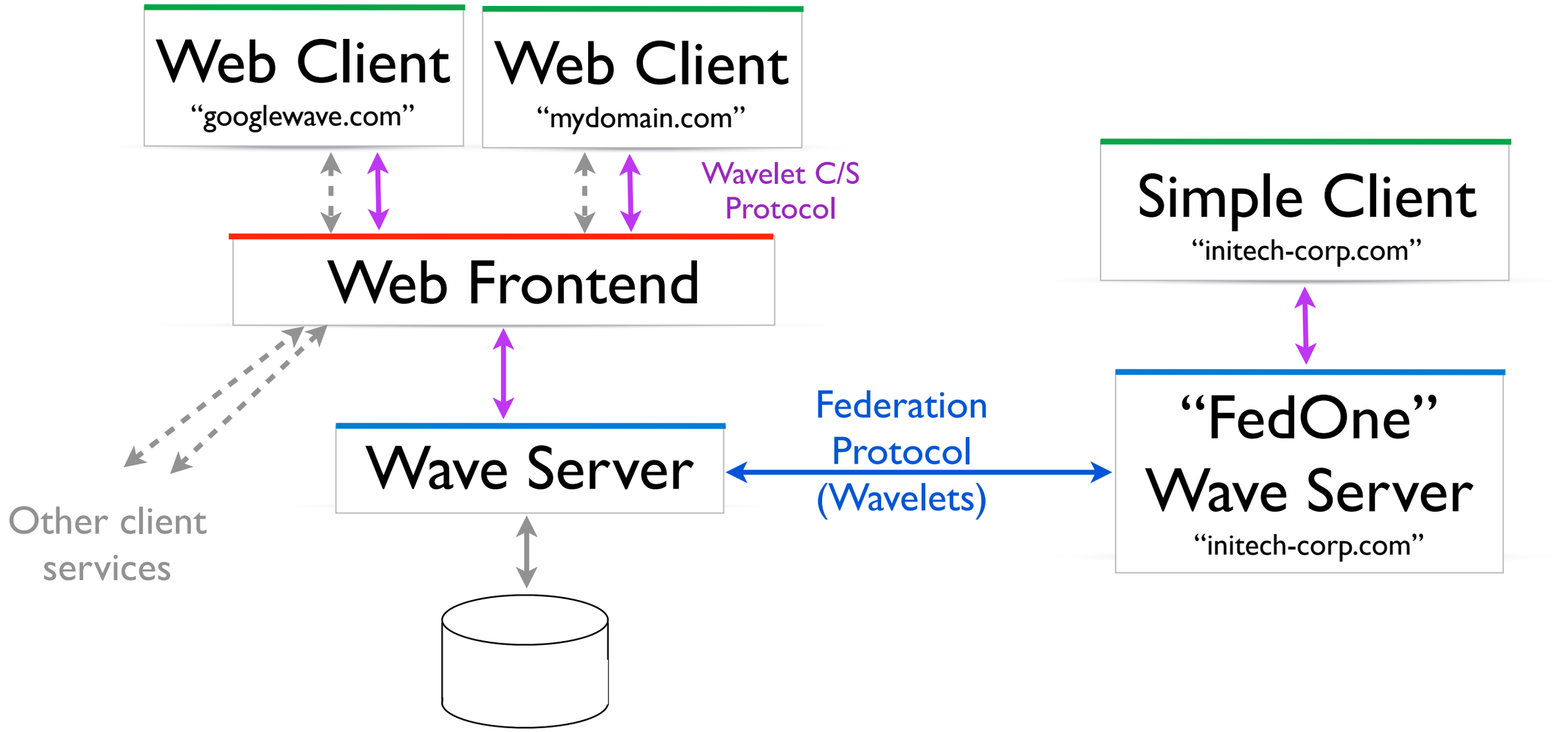


# Key Concepts

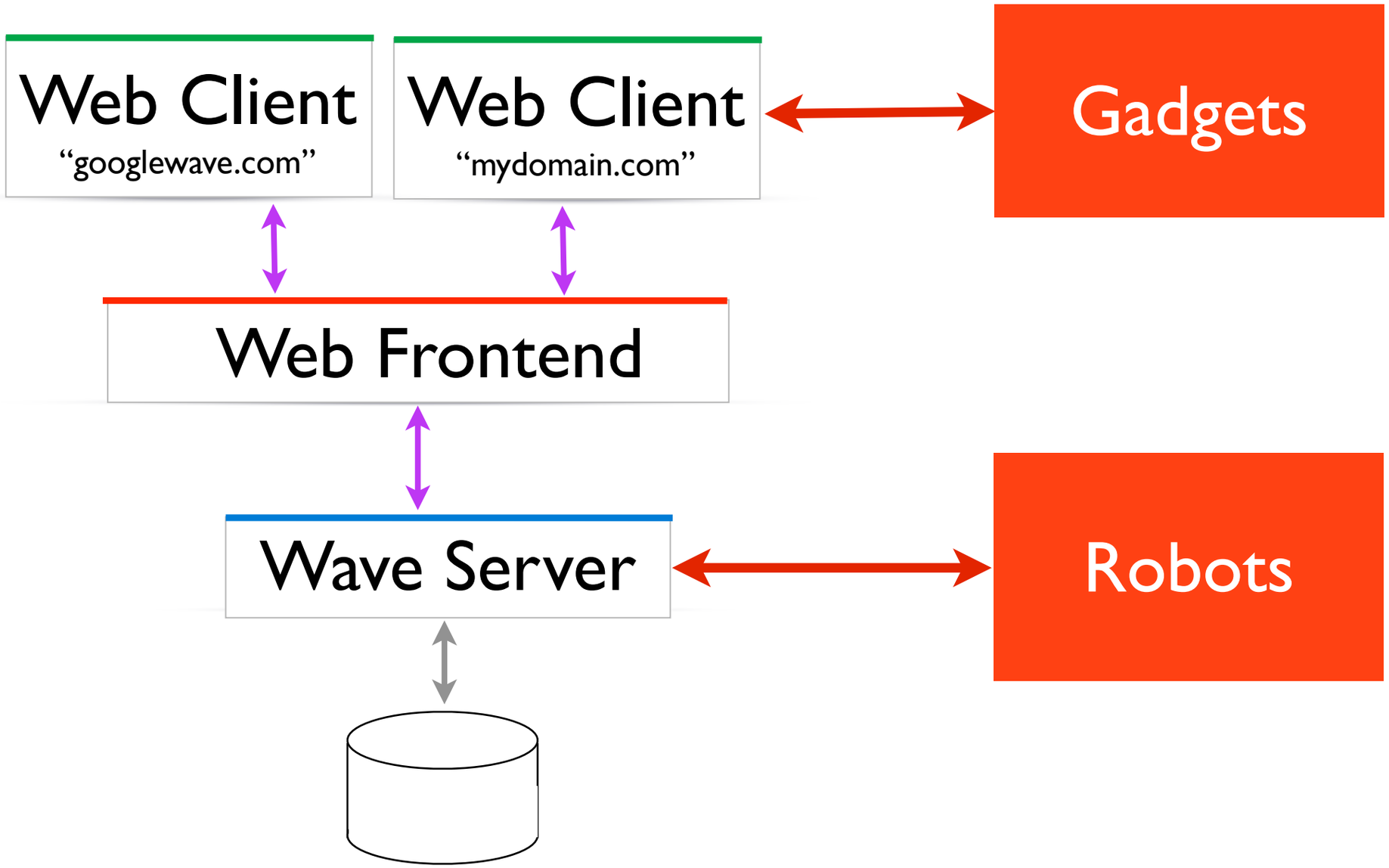



Also see: Operational Transformation http://bit.ly/xmzWO

# Overview Of Wave Server Architecture



# Write Extensions and Robots using the Wave API

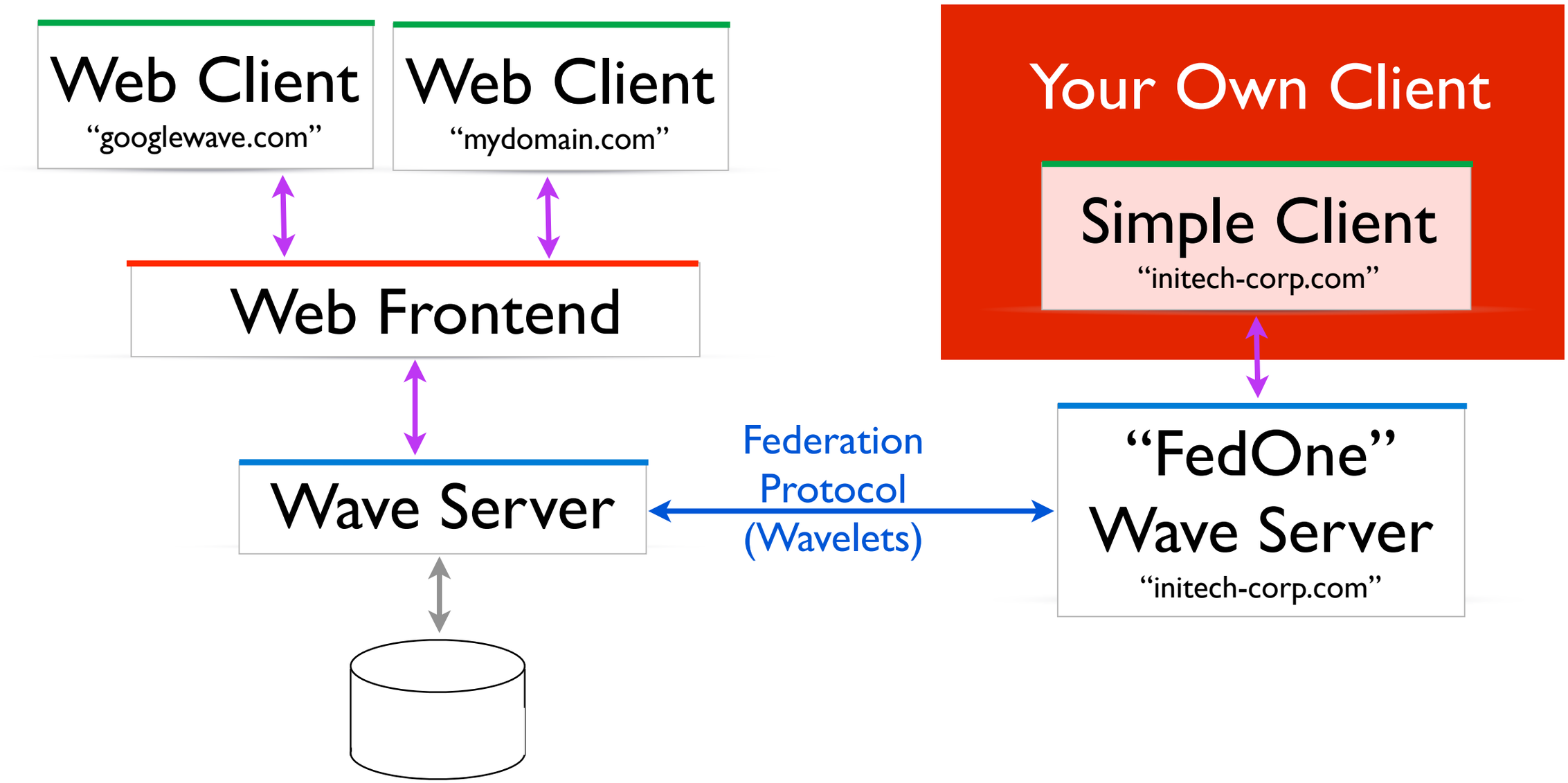



I/O Talks

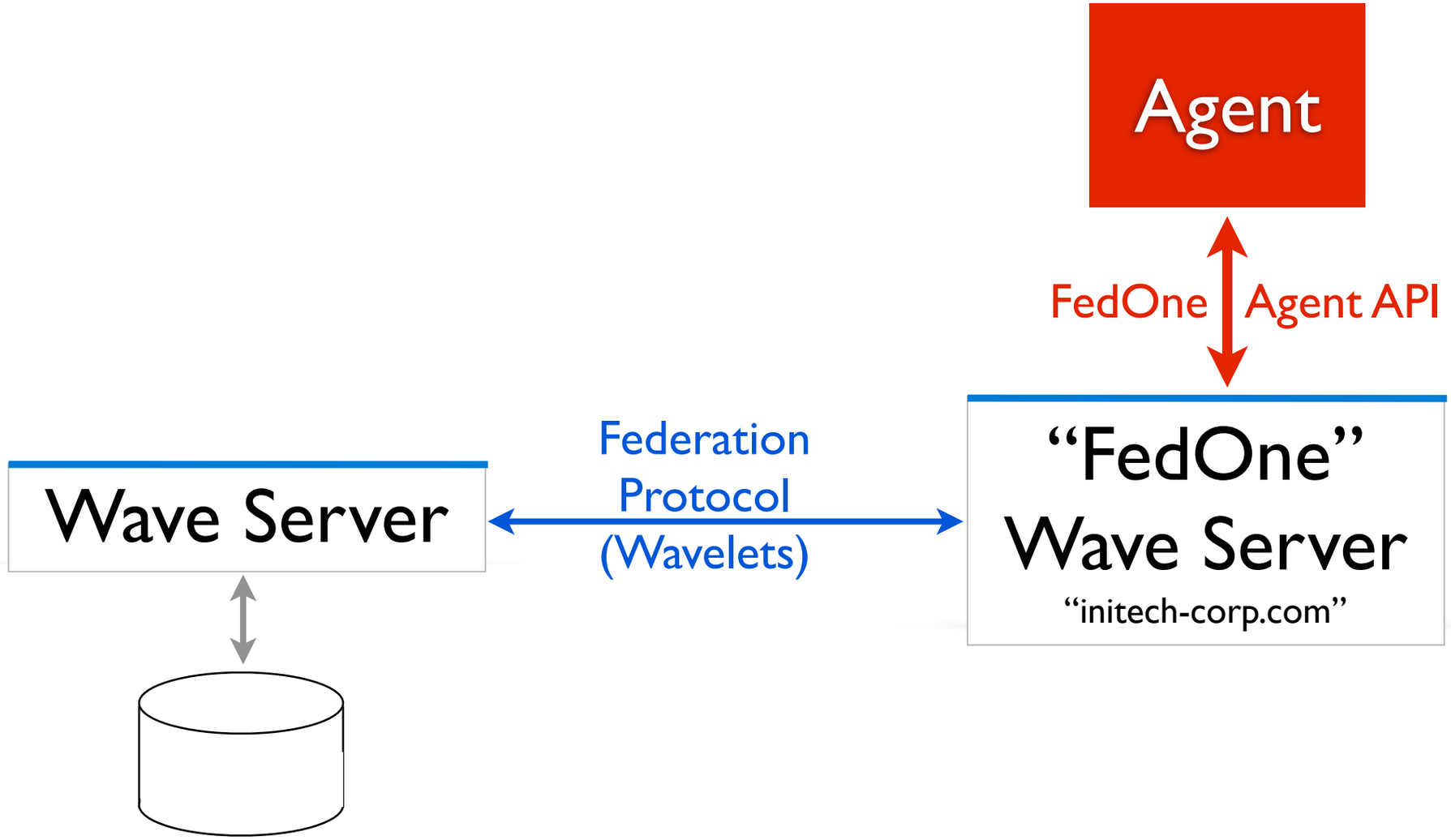
Google Wave API design principles: Anatomy of a great extension (held this morning)

Making smart & scalable Wave robots (Thursday May 20, 11:30am-12:30pm, Room: 8)

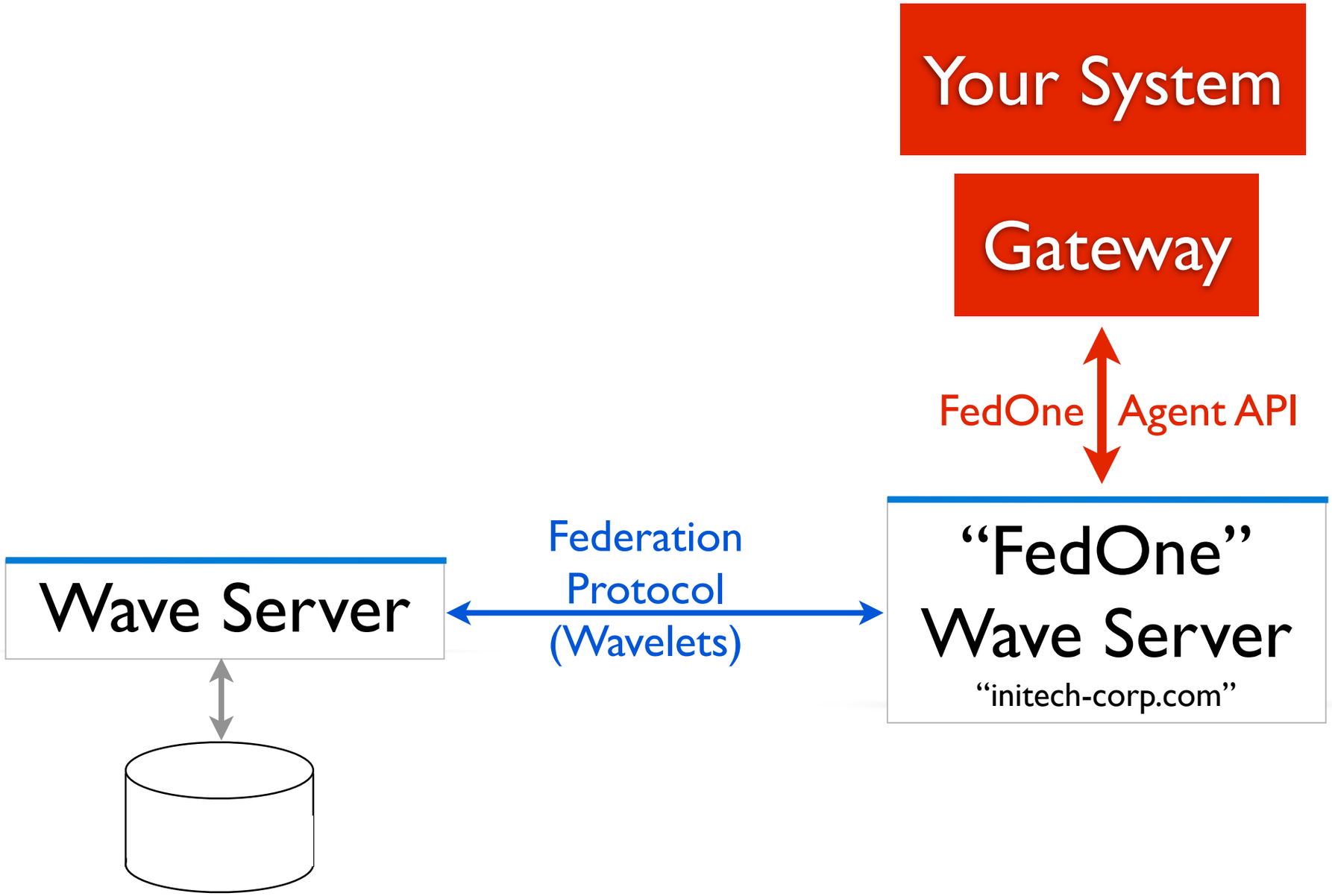
# Write your own client, or modify the Simple Client.



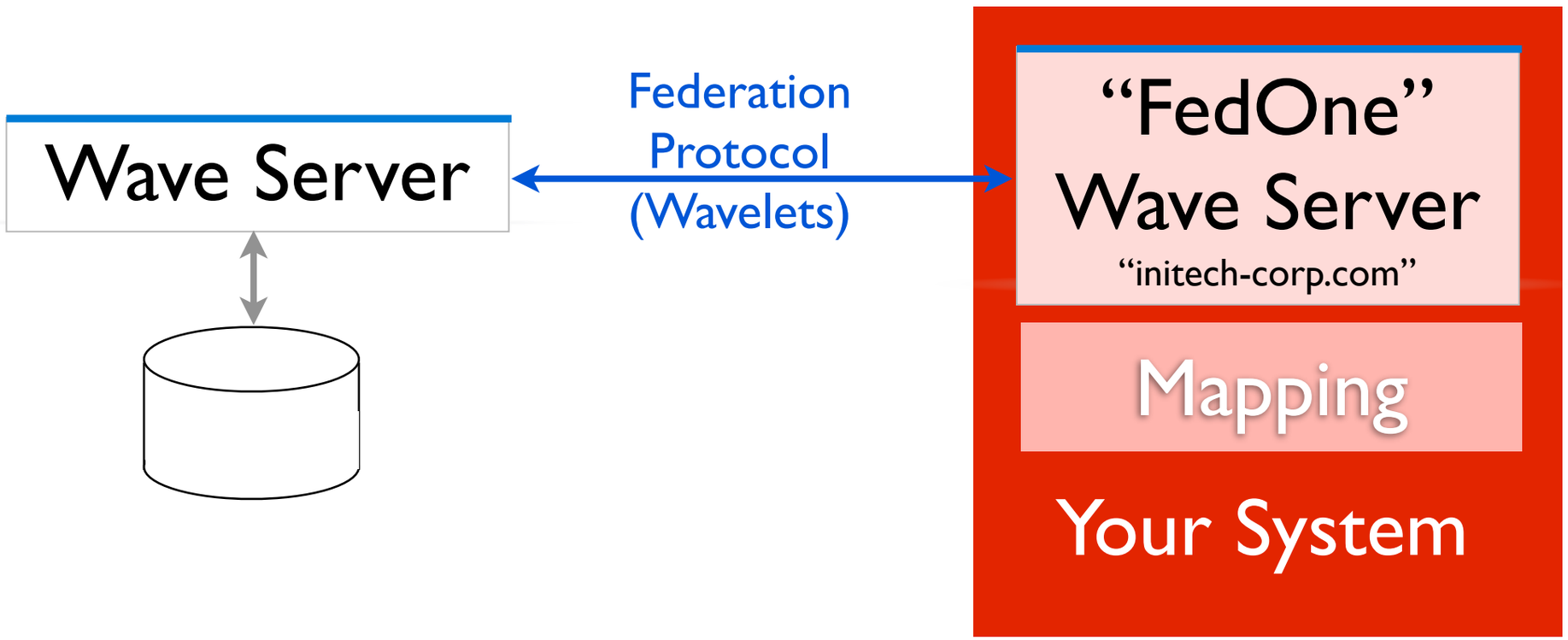
# Use the fledgling Agent API on FedOne



# Write FedOne Agent API Gateway



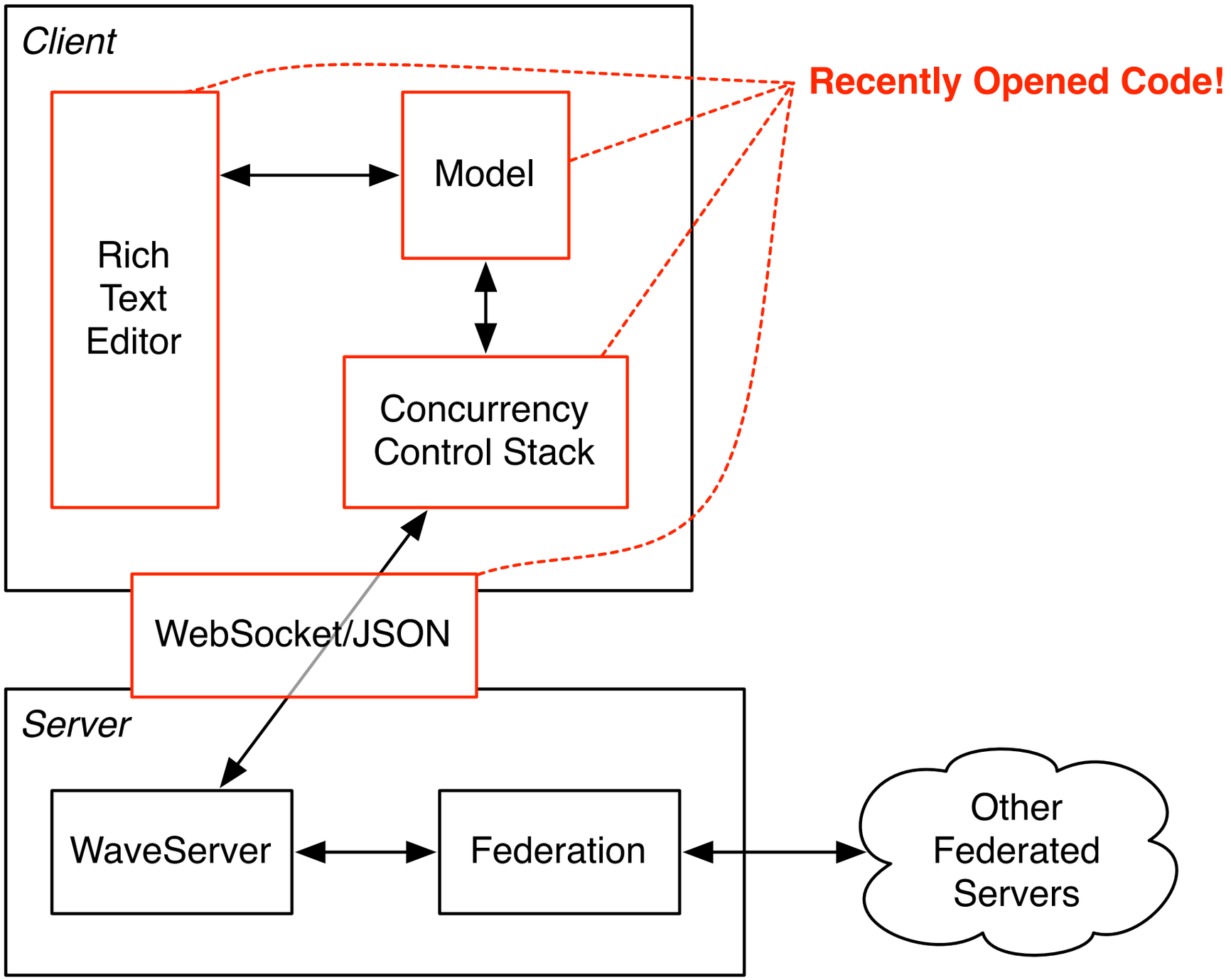
# Embed or Extend the FedOne Wave Server



# The Wave Model & Open Sourced Code

Google 10 I/O

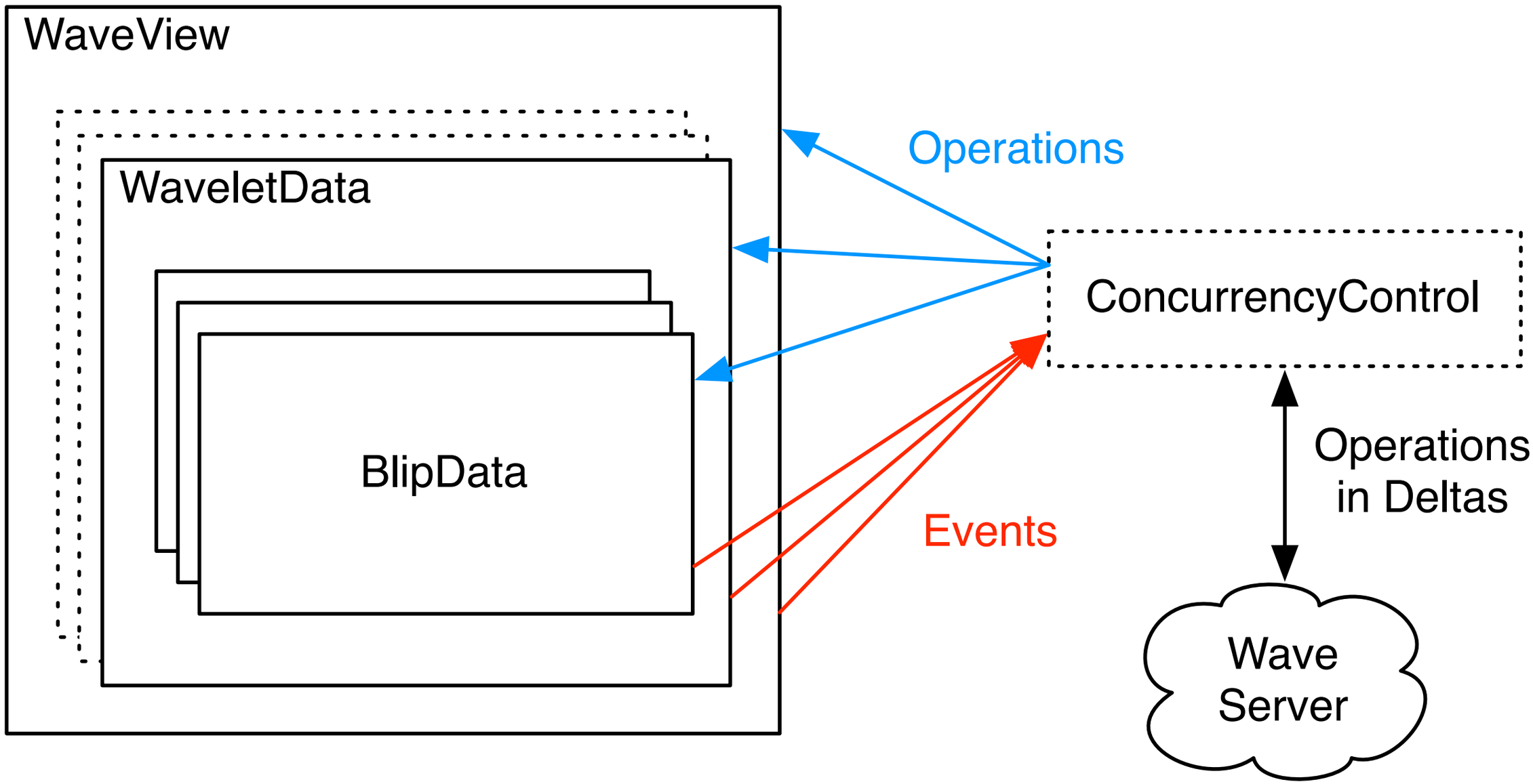
# Open Source Code



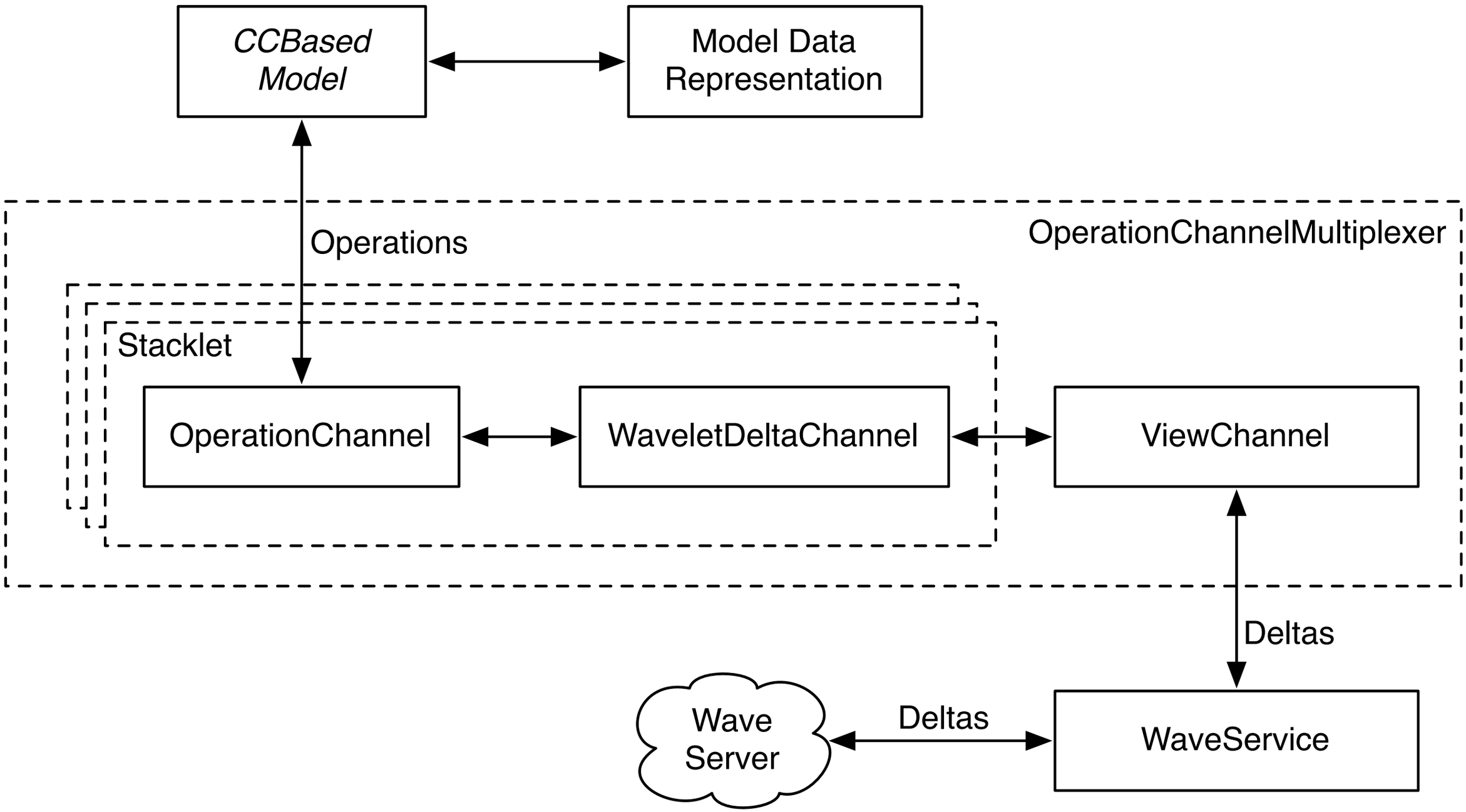
# New Open Source Code

- Major components that we are releasing today:
  - Wave Model
    - Document model
    - Conversation model
    - Wavelets & blips
    - Operations on documents and waves
  - Concurrency Control stack
    - Channels for communicating with Wave servers
    - Client implementations for flow control
    - Wave interface implementations that receive operations
  - Rich Text Editor

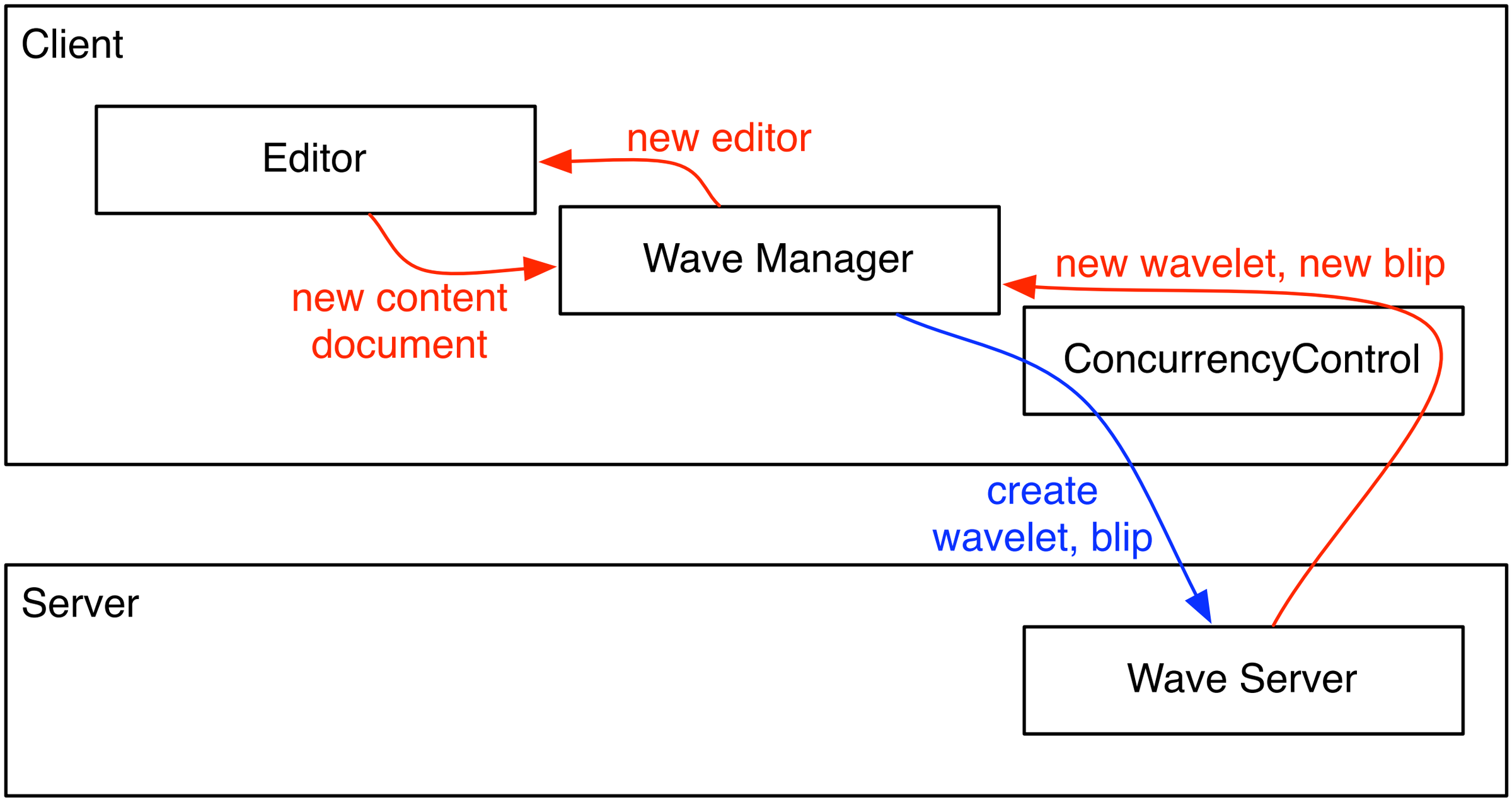
# Model Overview



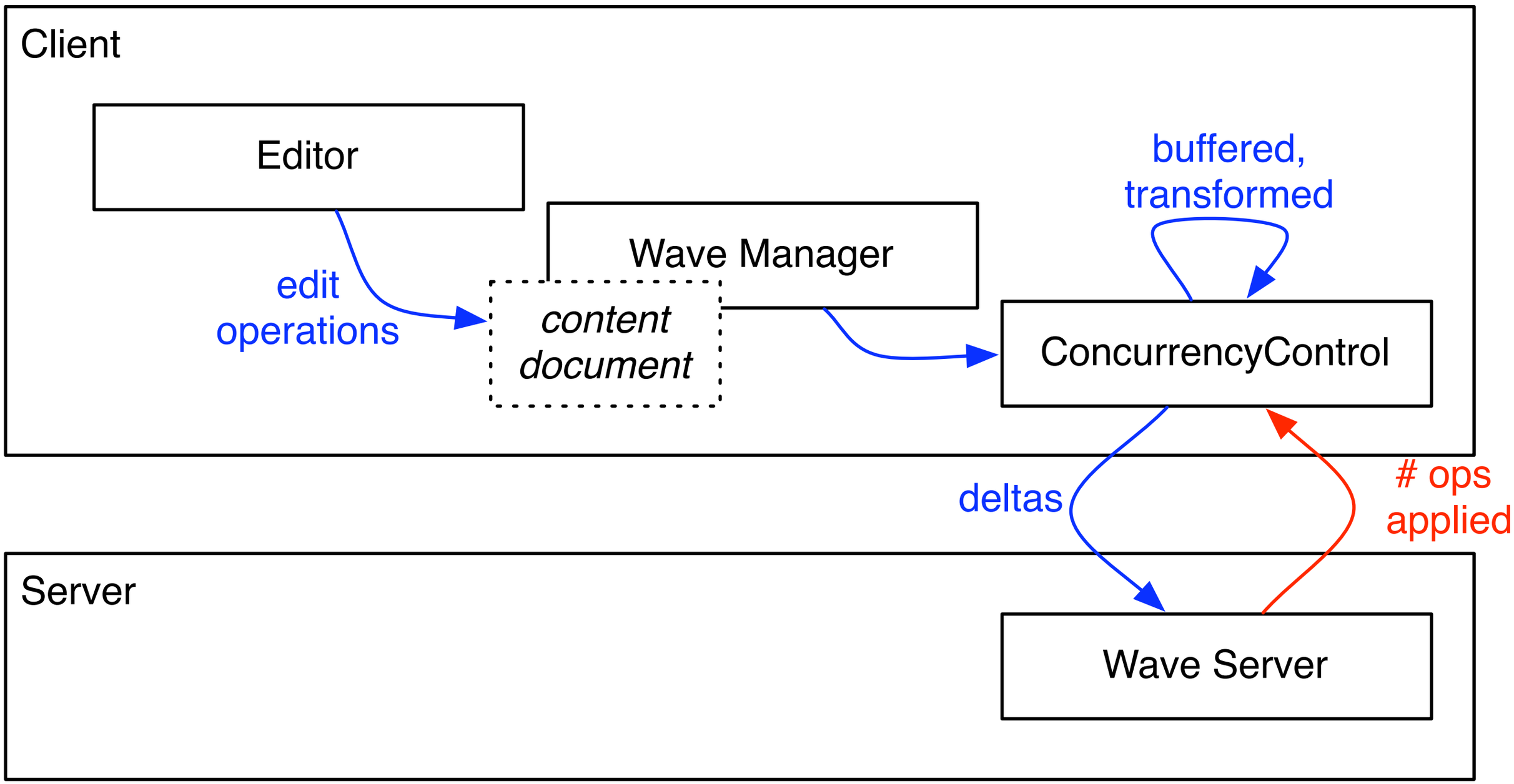
# Concurrency Control Overview



# Data flow (open / create)



# Data Flow, submit



# The Federation Protocol

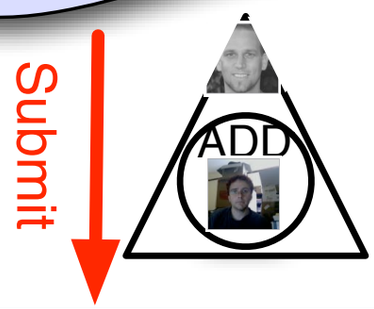
Google I/O 10

# The Google Wave Federation Protocol

- **Server to server protocol.**
- **Run your own on-premise wave service.**
- **Roll your own wave-based technology.**
- Draft Federation Protocol Spec: http://waveprotocol.org/
- Built on top of XMPP.
- Open sourced in July 2009.
- Same OT & CC for low-latency submit updates as client server protocol.

# Web Client

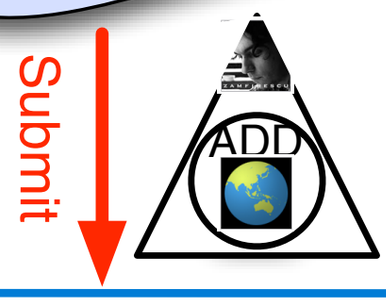
"acmewave.com"


# Wave Server


# Simple Client

"initech-corp.com"

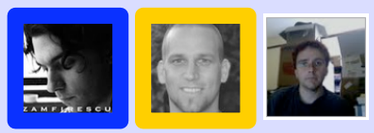

# "FedOne" Wave Server

"initech-corp.com"

Web Client

"acmewave.com"

"initech-corp.com!w+xcvS4"



Submit


Wave Server

"initech-corp.com!w+xcvS4"

copy only

Update



X.509 Signature

SHA256 [160 bits] History Hash

Simple Client

"initech-corp.com"

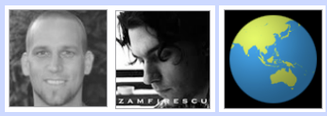
"initech-corp.com!w+xcvS4"

Submit

"FedOne" Wave Server

"initech-corp.com"


"initech-corp.com!w+xcvS4"

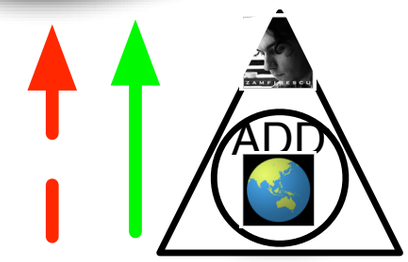


authorative

Web Client

"acmewave.com"

"initech-corp.com!w+xcvS4"




Update


Simple Client

"initech-corp.com"

"initech-corp.com!w+xcvS4"



Wave Server

"FedOne" Wave Server

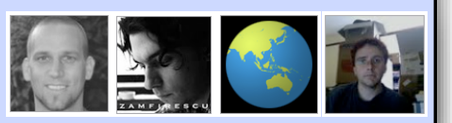
"initech-corp.com"

Ack

"initech-corp.com!w+xcvS4"

copy only

"initech-corp.com!w+xcvS4"



authorative

# 4 Basic Messages

- Submit Request / Response
- Updates
- History Request / Response
- Signer Information Request / Response

# Top Features Needed to Complete the WFP

- Reliable Delivery.
- Federated Attachments.
- Federated Groups.
- Federated Presence.

# Further Documentation for WFP

- http://waveprotocol.org/
- http://code.google.com/p/wave-protocol/
- http://code.google.com/p/wave-protocol/wiki/FurtherReading
- Shout-out: Anthony Watkins, James Purser, Bryce

# Client / Server Protocol

# Client-Server Protocol

- Protocol for receiving updates and submitting changes to a Wave server directly
- WebSocket transport, JSON-encoded protobuffer-style messages
  - *protobuffer-style:* keyed by number

# Client-Server Protocol, sending updates

- Combine outstanding operations into a **ProtocolWaveletDelta**
- Put delta into **ProtocolSubmitRequest**, and submit it.

- The **ProtocolSubmitResponse** has operation count and error message

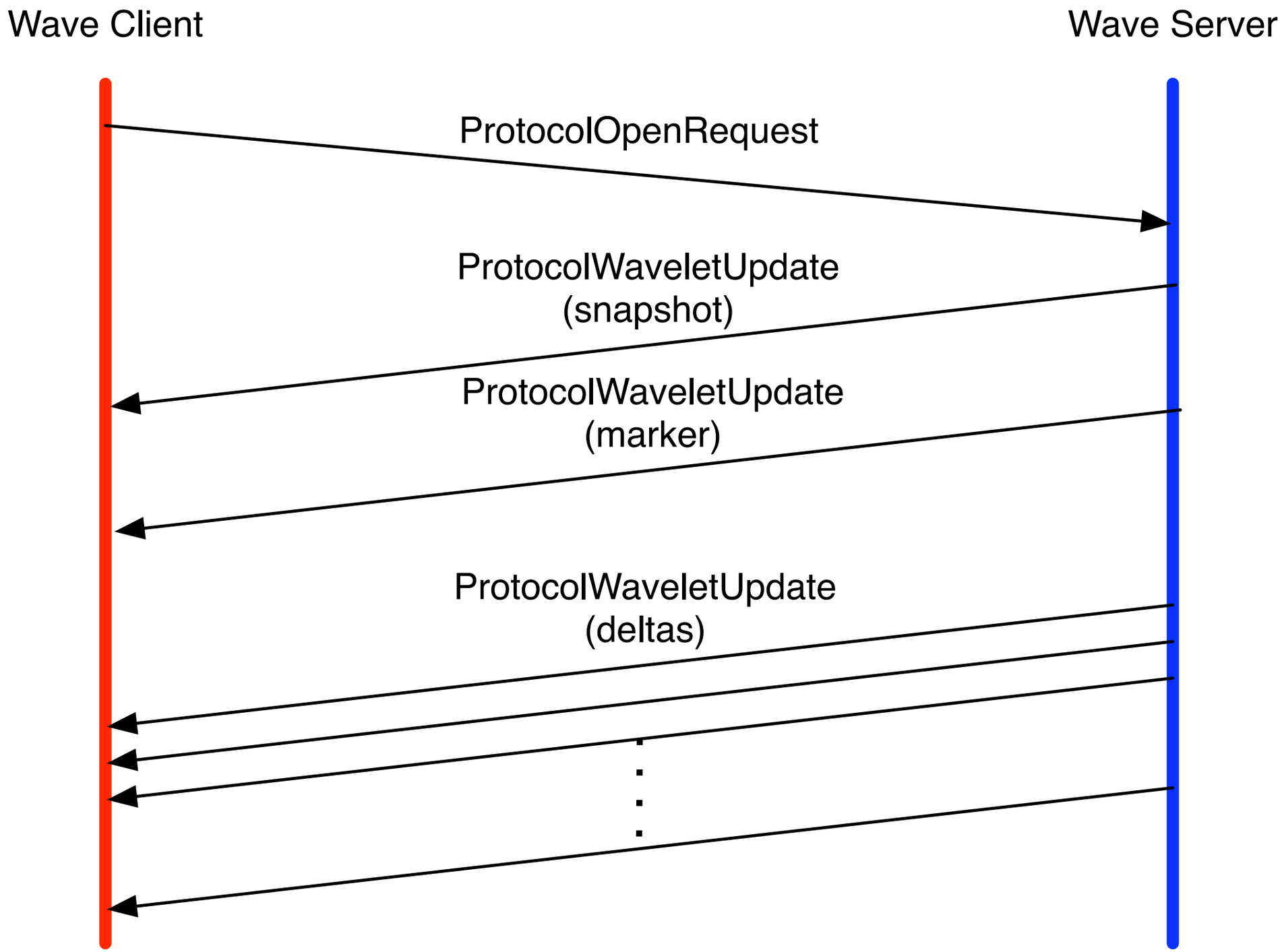
# Client-Server Protocol, receiving updates

- Send **ProtocolOpenRequest**
  - WaveId
  - WaveletId prefix
  - Known snapshot versions

- Receive **ProtocolWaveletUpdate**
  - Snapshots for requested wavelets

- Receive **ProtocolWaveletUpdate***
  - Updates (deltas) for requested wavelets, transform them against locally outstanding operations, and apply

# Client-Server Protocol, Receiving Updates



# Demos

# Emacs Client

Google I/O 10

# SAP and Google Wave Federating

Google™ I/O 10

# Novell and Google Wave Federating

Google™ I/O 10

# SAP, Novell, and Google Wave Federating

Google I/O 10

# Conclusion

# Google Wave Federation Protocol

Get involved: http://waveprotocol.org

Google I/O 10

# Thank you