

**CALCONNECT DOCUMENT CD 0801**

**Type:** Presentation  
**Title:** CalConnect Technical Preview - Roundtable XI  
**Version:** 1.0  
**Date:** 2008-02-06  
**Status:** Published  
**Source:** N/A

This document incorporates by reference the CalConnect Intellectual Property Rights, Appropriate Usage, Trademarks and Disclaimer of Warranty for External (Public) Documents as located at

<http://www.calconnect.org/documents/disclaimerpublic.pdf>.

# CALCONNECT

The Calendaring and Scheduling Consortium

## Technical Preview

February 2008 Roundtable

# Note Well

- This presentation is derived from the Technology Preview material
- Names and some technical details have been removed in accordance with CalConnect rules involving in-progress work
- This is a public document and may be quoted and distributed in accordance with CalConnect policies

# Introduction

CalConnect members have produced a Technical Demonstration of key calendaring and scheduling technologies being developed by CalConnect

This serves as a technology “preview” only and does not in any way represent final protocols or products that may or may not ship

More...

# Introduction (cont.)

- This is a follow-up to the demo at Roundtable X (September 2007)
- Our goal is to solicit feedback from members and invited guests on the presentation itself as well as the technologies being demonstrated
- We also hope to show how CalConnect is successfully achieving its goal of improving calendaring and scheduling standards

# Agenda

- Today we will demonstrate:
  - CalDAV Scheduling
  - Realtime Internet Scheduling
  - Internet freebusy lookups using freebusy URLs
- Each presentation will consist of:
  - Introductory slides
  - Live demonstration

# Technologies

- CalDAV Scheduling:
    - The CalDAV access protocol (RFC4791) is based on WebDAV and iCalendar standards
    - In the CalDAV scheduling specification (still a draft), users on the same server system only schedule with each other
    - There are several server and client implementations available
- More...

# Technologies (cont.)

- Realtime Internet Scheduling:
  - Allows scheduling between users on different server systems (e.g. between organizations)
  - The protocol uses HTTP to transport iCalendar scheduling messages
  - We will be demonstrating the basic exchange of scheduling messages
  - The security and discovery pieces are still to do (TC-REALTIME)

More...



# Technologies (cont.)

- Freebusy URL:
  - This is a way to do simple exchange and publishing of freebusy information
  - TC-FREEBUSY is working on a specification
  - It uses simple HTTP GET/PUT operations with discoverability

# CalDAV Scheduling

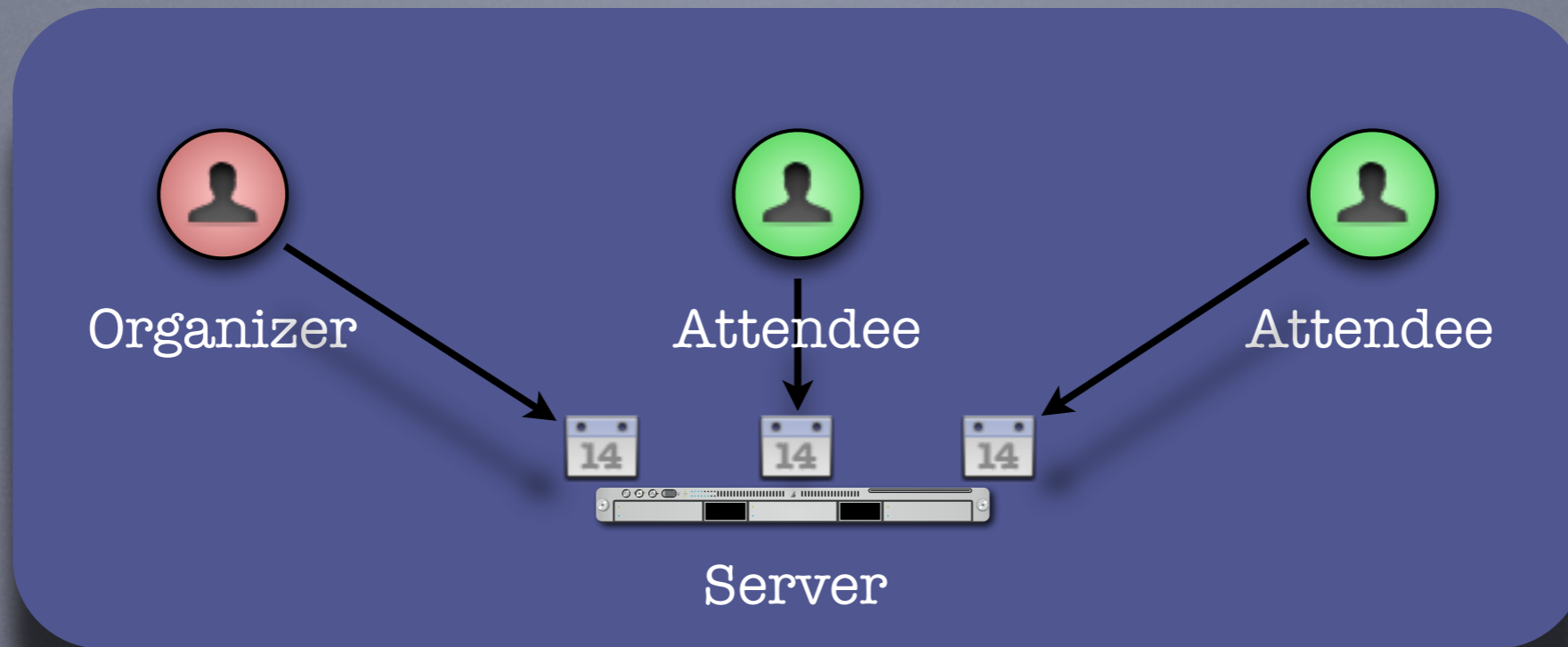
How it works

# CalDAV Scheduling

- How it works:
  - Several users on one CalDAV server (can be using different clients) schedule with each other
  - One user is the “Organizer”, others are “Attendees”

# One is the Organizer, others are Attendees

Each stores their calendars on the Server



# CalDAV Scheduling

- There are two parts to scheduling:
  - Freebusy lookup
  - Sending invitations and receiving replies
- Freebusy results are returned immediately
- Invitation replies are sent only after users inspect and accept or decline

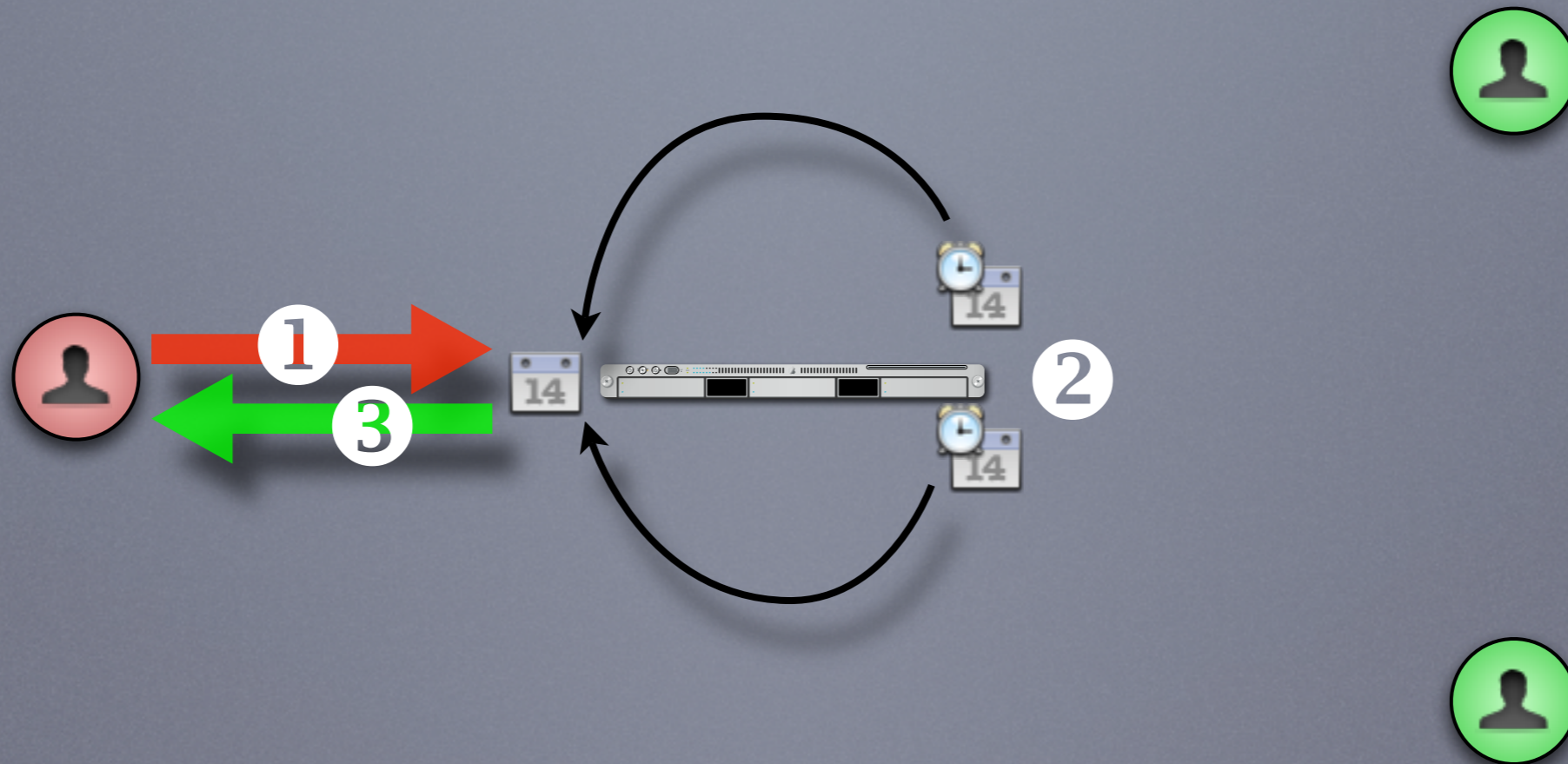
# CalDAV Scheduling

- Each user has an “Outbox” and an “Inbox”
- The “Outbox” is used to trigger freebusy lookup or send invites or replies
- The “Inbox” is where invites or replies are delivered
- Clients monitor the “Inbox” for incoming scheduling messages

# CalDAV Scheduling

Sequence of diagrams showing free-busy  
lookup.

- 1 Organizer sends freebusy request to server
- 2 Server calculates freebusy data for attendees
- 3 Server returns aggregated freebusy data to organizer



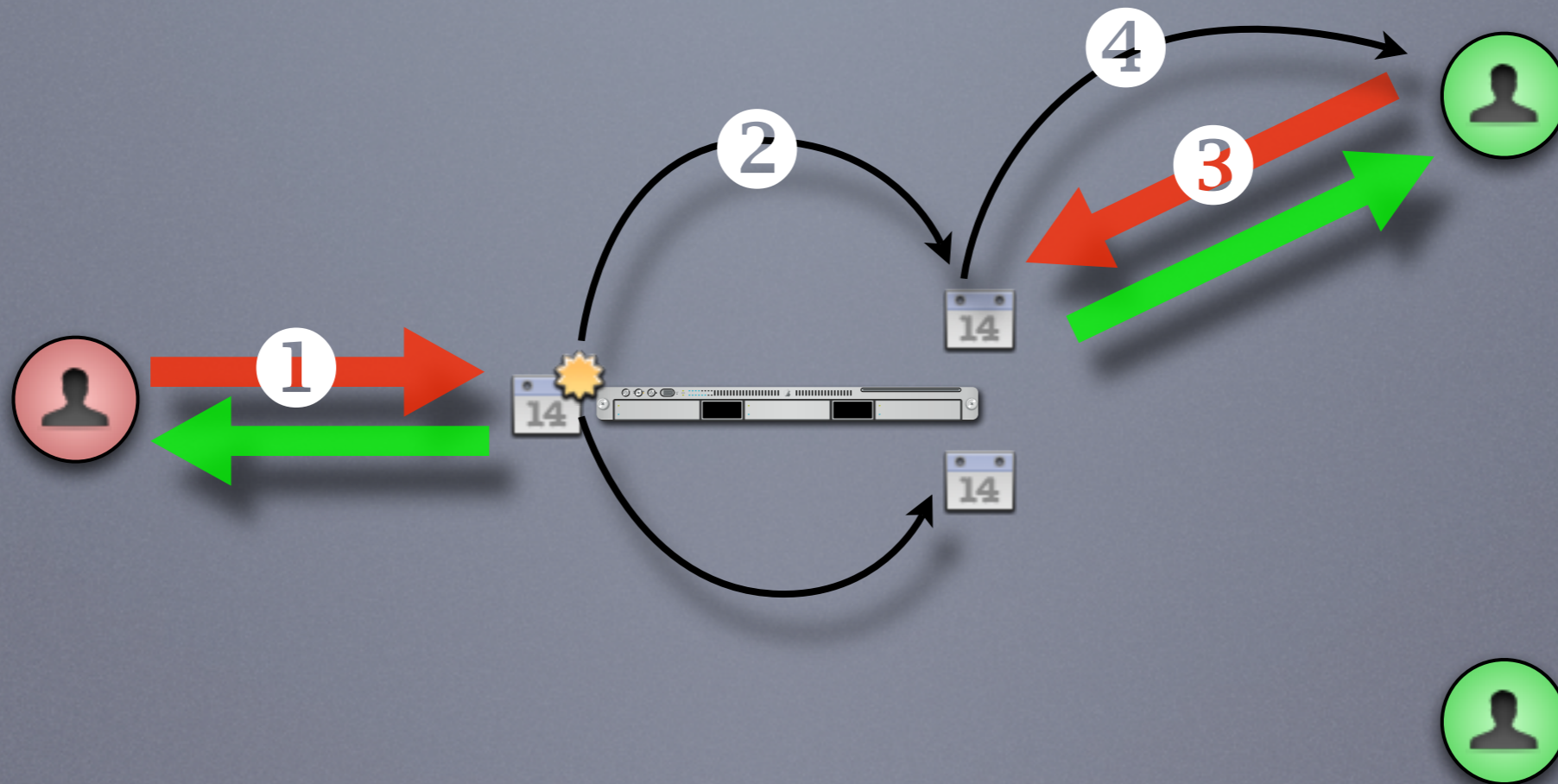




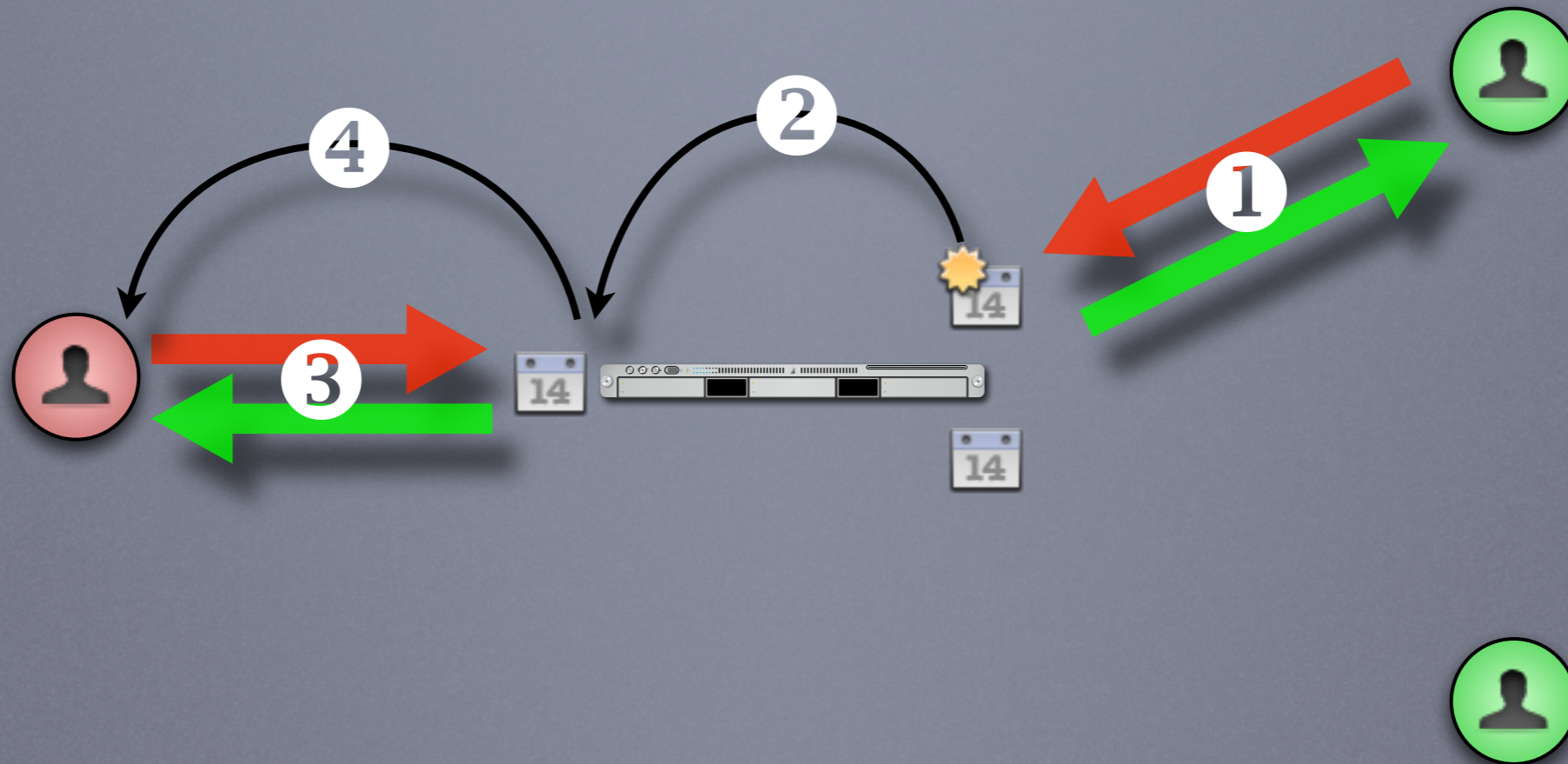
# CalDAV Scheduling

Sequence of diagrams showing invitations being sent out, replies returned.

- ① Organizer sends invite request to server
- ② Server copies the request into attendees' Inbox
- ③ Attendee checks the server
- ④ Attendee retrieves the invite from the server



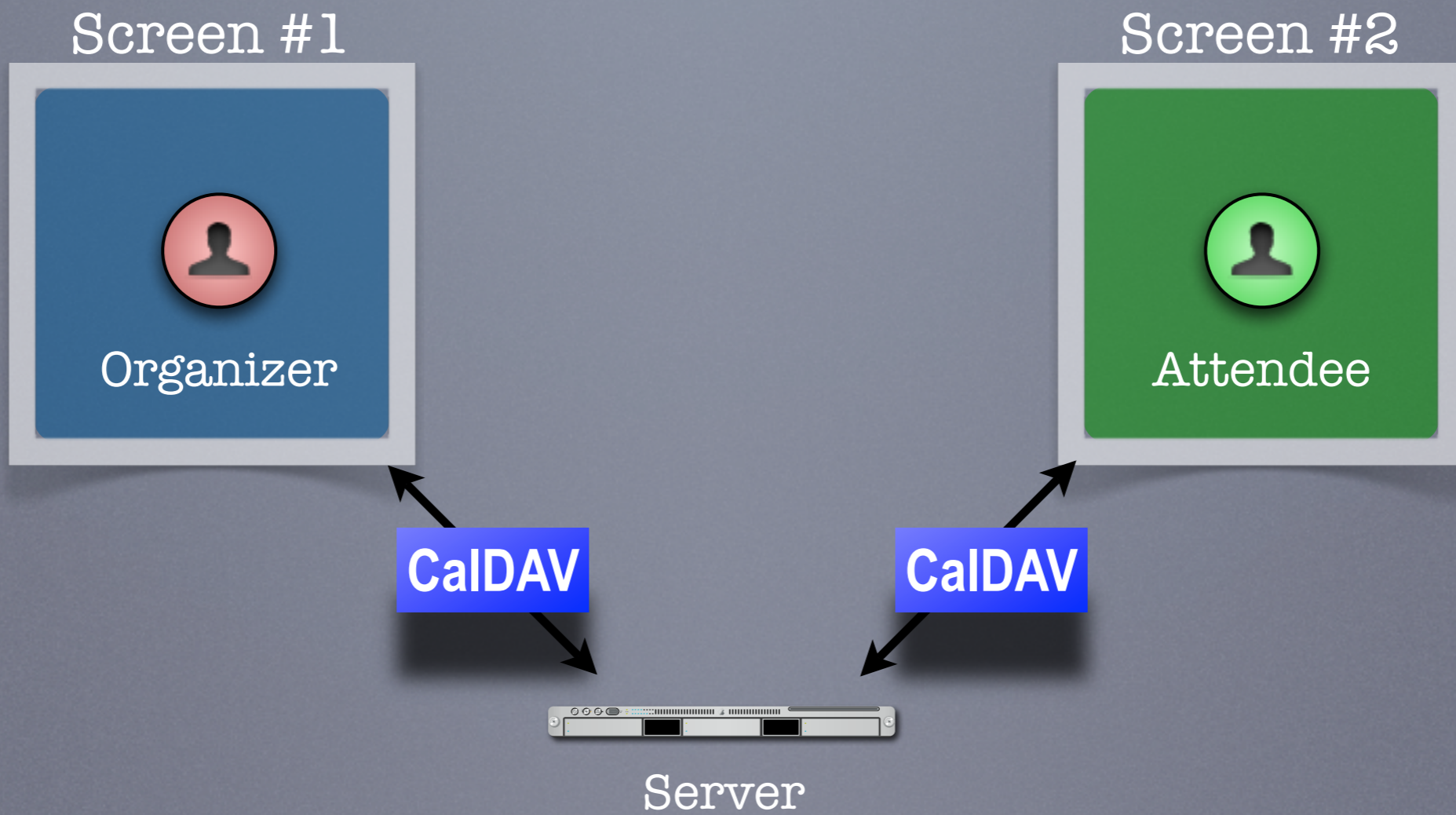
- ① Attendee replies to the server
- ② Server copies the reply into organizer's Inbox
- ③ Organizer checks the server
- ④ Organizer retrieves the reply from the server



# CalDAV Scheduling Demonstration #1

Simple meeting between two people

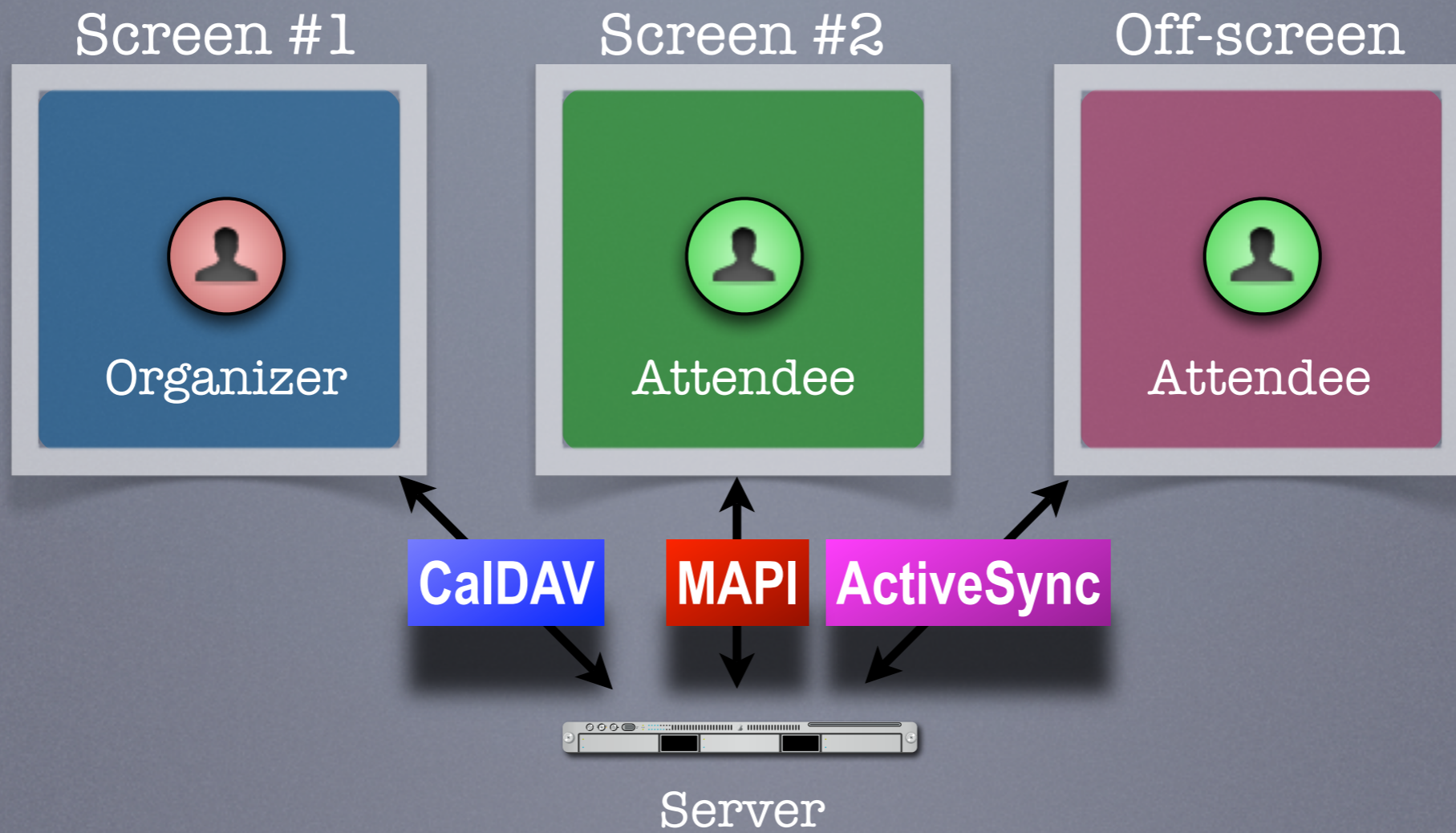
# Demo Participants



# CalDAV Scheduling Demonstration #2

Simple meeting between multiple people  
with different clients some CalDAV others  
using a CalDAV “connector”

# Demo Participants





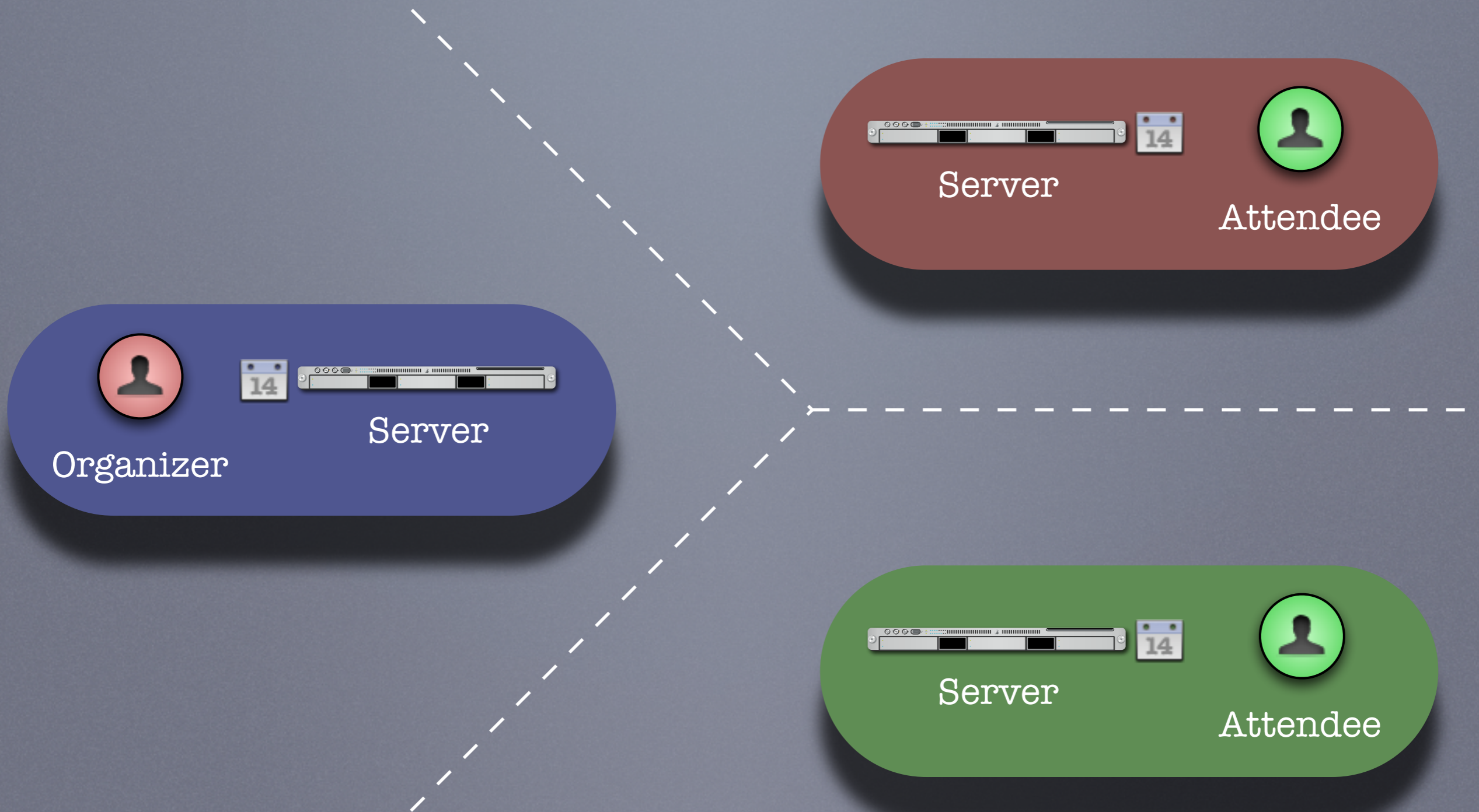
# Realtime Internet Scheduling

How it works

# Basic Concept

- Provides the ability for users on different calendaring systems to schedule meetings with each other
- Instantaneous freebusy lookups
- Invites, replies sent as “messages” with delivery status immediately returned

# Organizer and Attendees on different systems



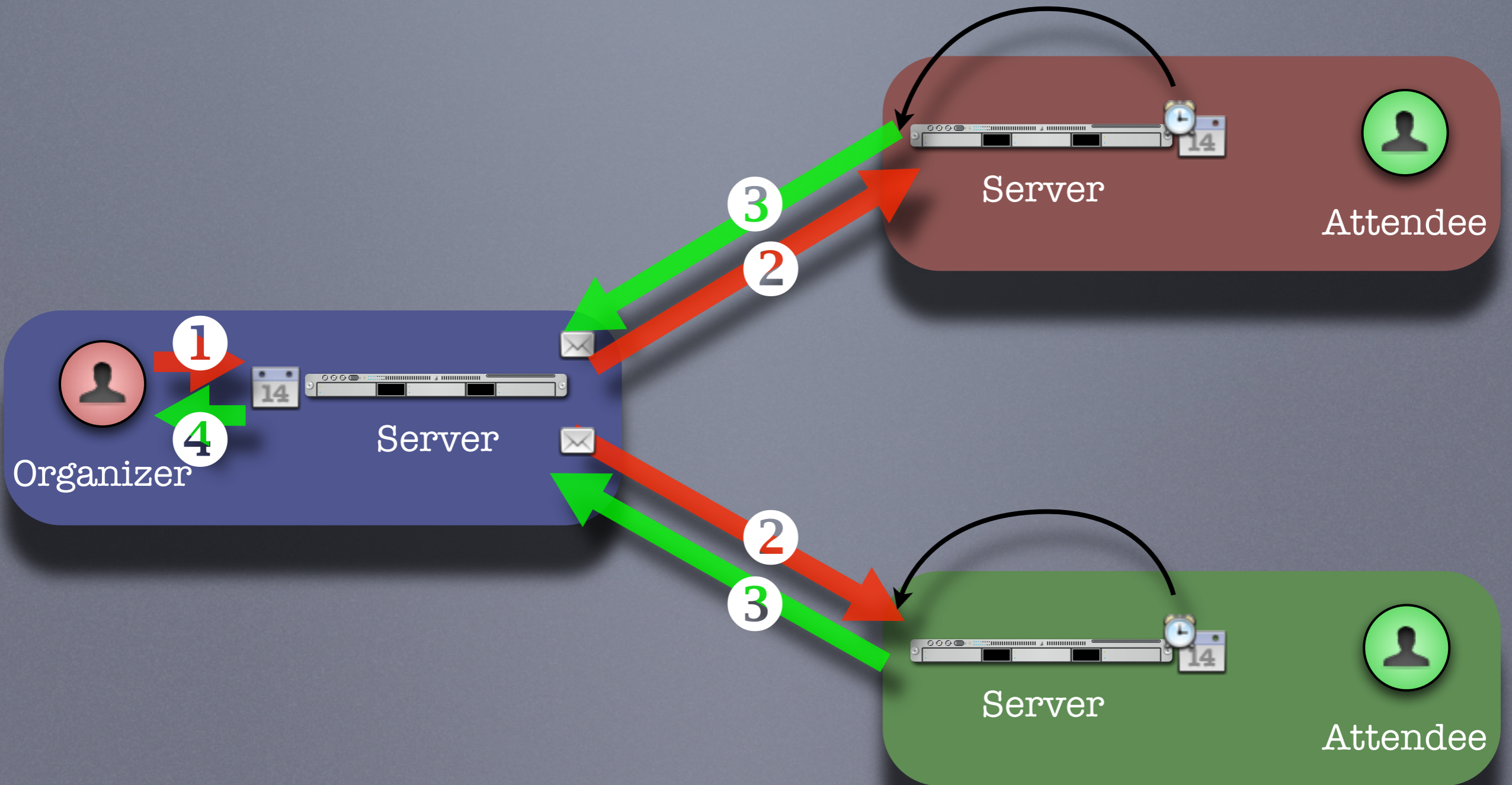
# Can't this be done today?

- But I can do scheduling with my colleagues today!
- True, but only people on the same server as you, or via some other communication process such as email or telephone.

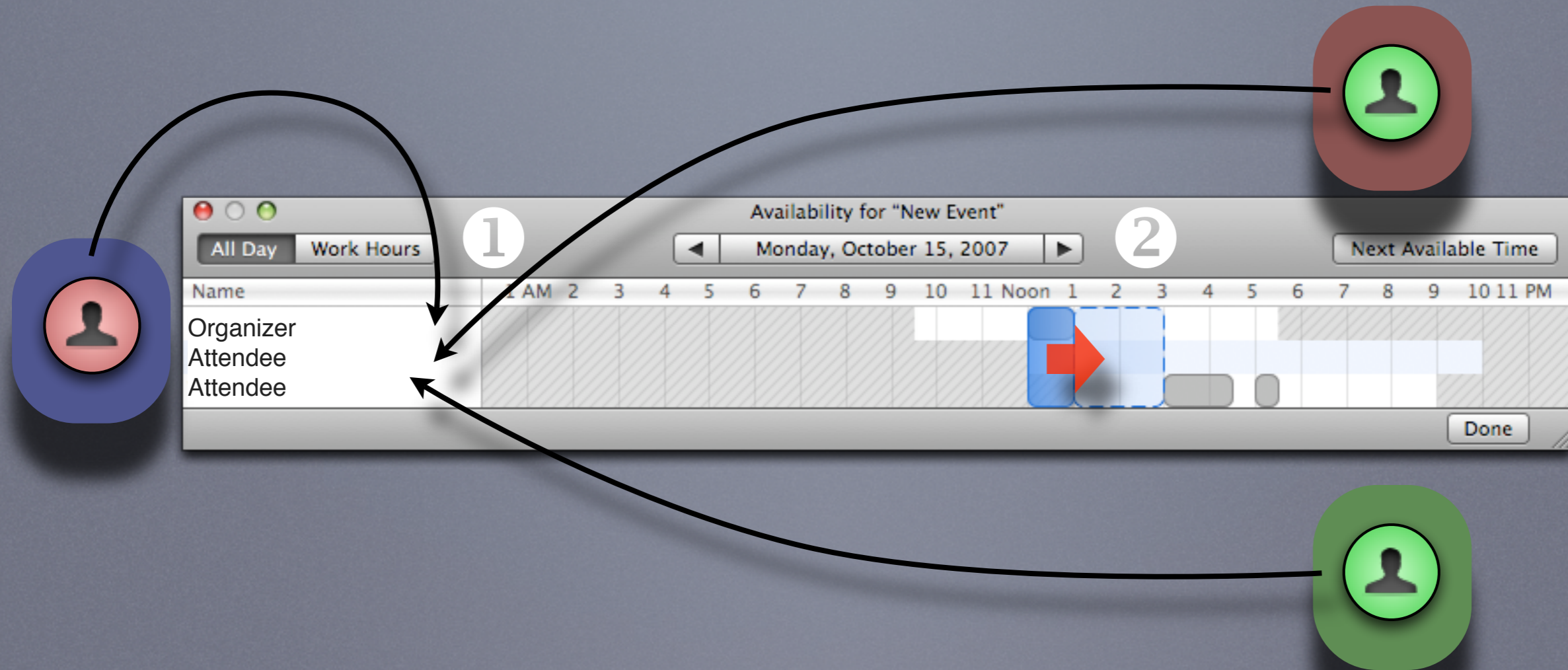
# Realtime Internet Scheduling

Sequence of diagrams showing freebusy.

- 1 Organizer sends freebusy request to server
- 2 Server forwards request to other attendee's servers
- 3 Attendees' servers return freebusy
- 4 Server returns aggregated freebusy data to organizer



- 1 Organizer sees freebusy for everyone
- 2 Can adjust meeting time to avoid conflicts

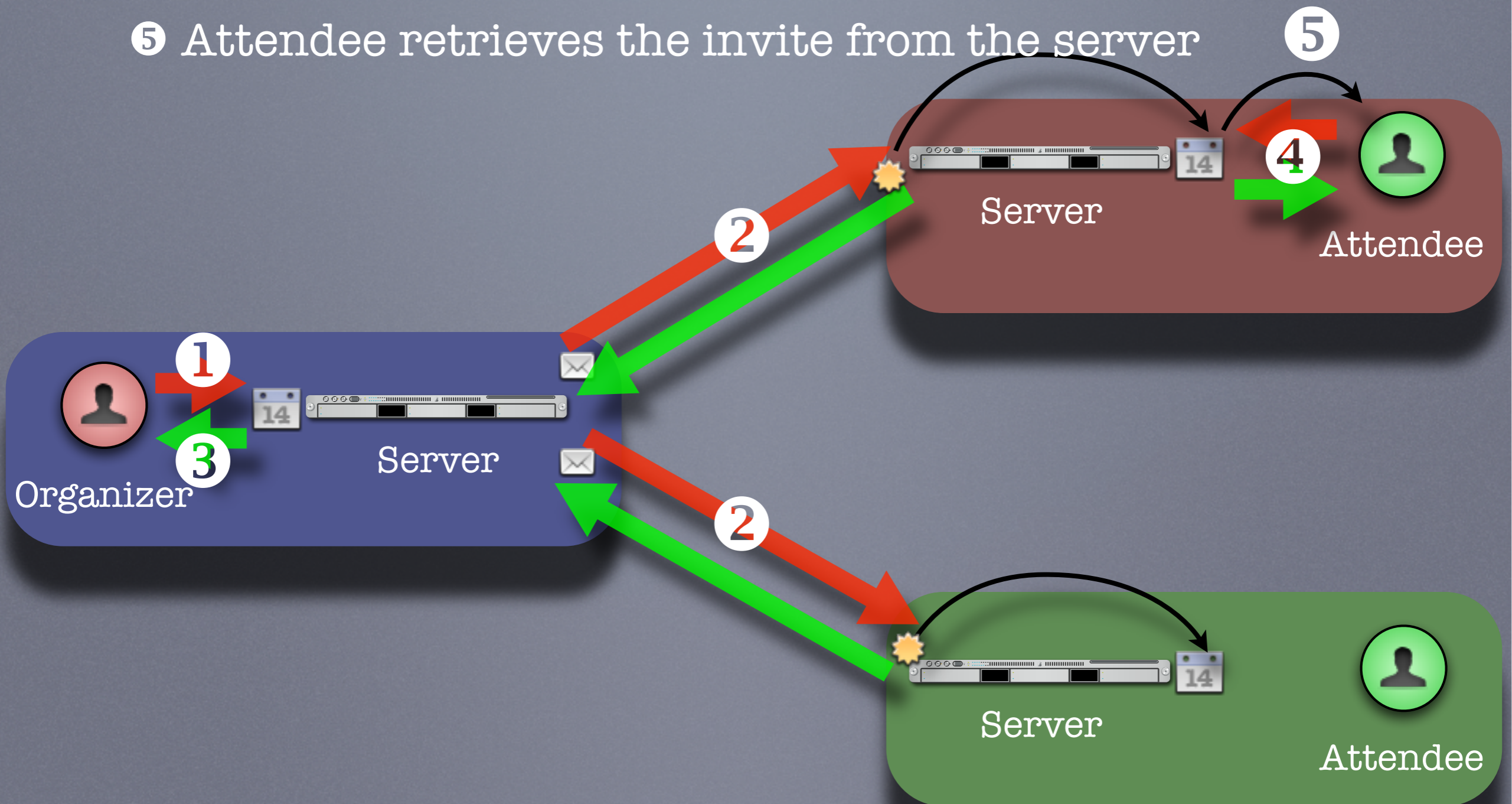


# Realtime Internet Scheduling

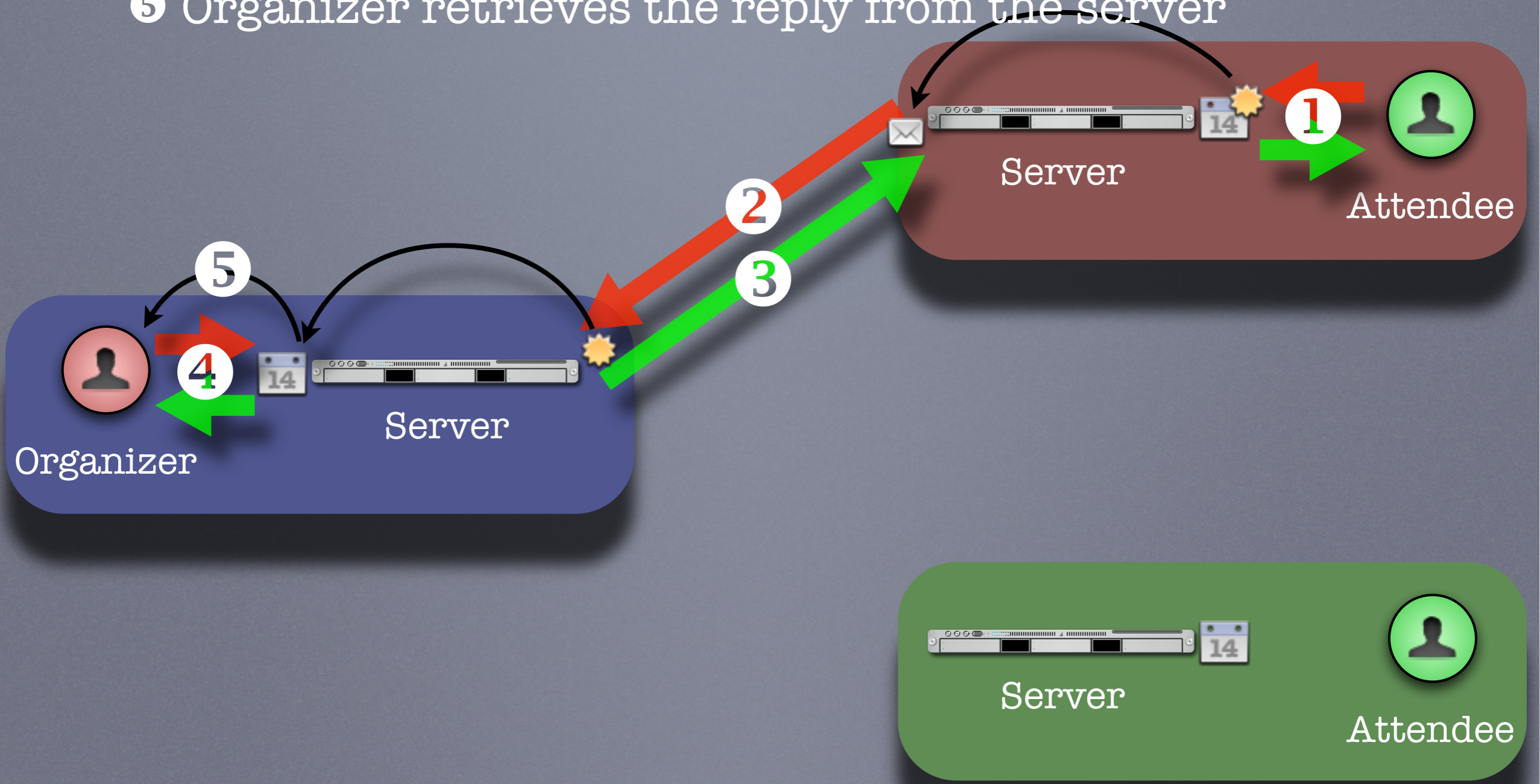
Sequence of diagrams showing invites  
and replies.



- 1 Organizer sends invite request to server
- 2 Server forwards request to other attendee's servers
- 3 Organizer gets delivery status response
- 4 Attendee checks the server
- 5 Attendee retrieves the invite from the server



- 1 Attendee sends reply to server
- 2 Server forwards reply to other organizer's server
- 3 Attendee gets delivery status response
- 4 Organizer checks the server
- 5 Organizer retrieves the reply from the server



# Realtime Internet Scheduling Demonstration

Four calendar users in different domains

# Realtime Demo Setup



Organizer

Screen #1

**CalDAV**



Server



Attendee

Screen #2

**CalDAV**



Server

**Realtime**

Screen #1



Organizer

**Web**



Server



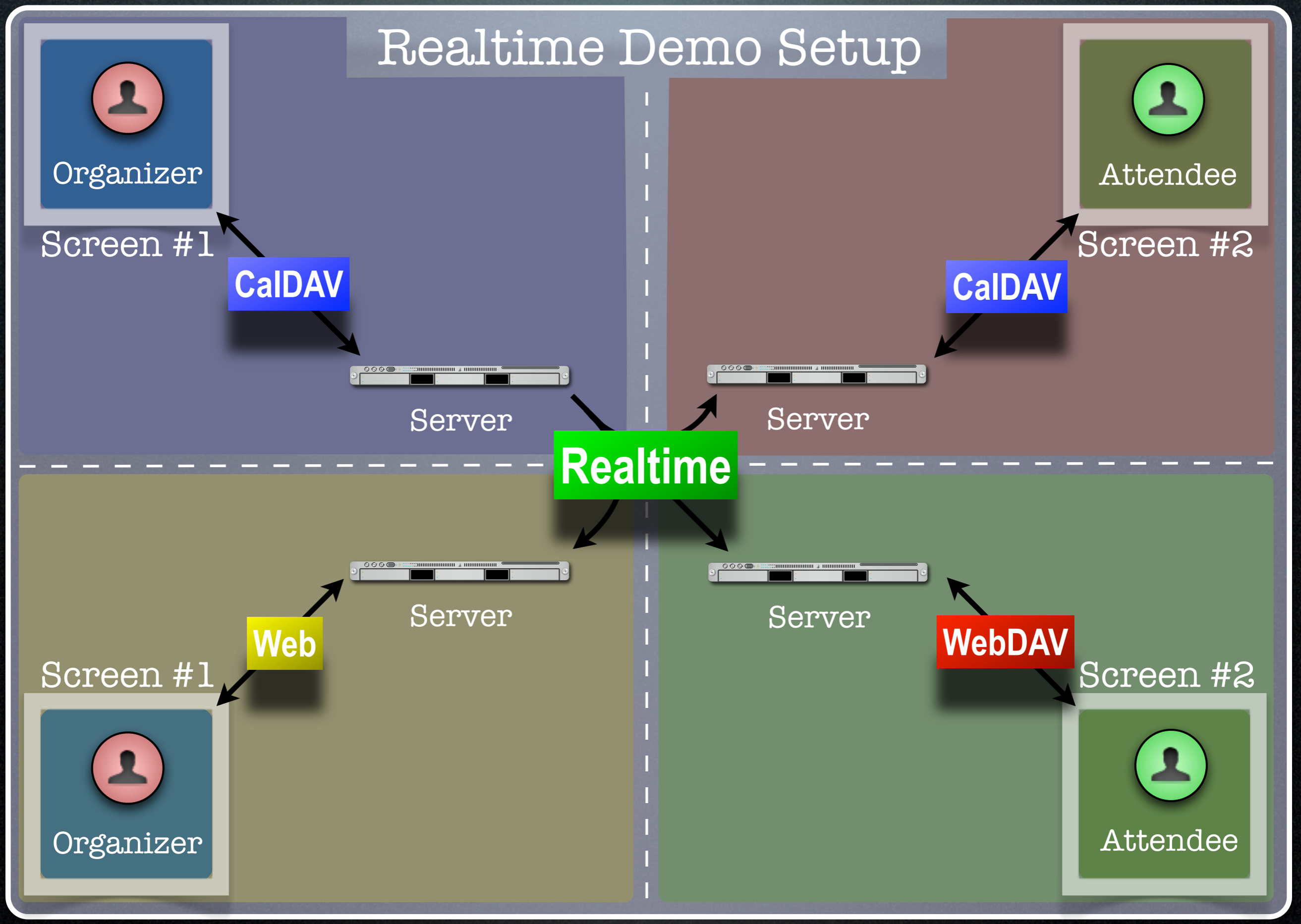
Server

**WebDAV**

Screen #2



Attendee



Freebusy URL

# What is Freebusy?

- A list of free and busy periods for a particular calendar user or resource
- Primarily used for scheduling resources or meetings with other people
- Time periods may be marked as
  - busy
  - free
  - busy unavailable ("out of office")
  - busy-tentative

# Expressing Freebusy time

- Most commonly as a RFC 2445 VFREEBUSY object
  - a request for freebusy time,
  - a response to a request, or
  - a published set of busy time

# Sharing Freebusy

- CalDAV Scheduling
- iTIP/iMIP (email)
- iCalendar .ics file
- Freebusy URL (FBURL)



# Why FBURL?

- Freebusy is “least common denominator” (LCD) scheduling
- FBURL is LCD Freebusy (or could be)
- Easy
- Outlook supports a form of FBURL

# Why FBURL? (cont.)

- The market says FBURL is desirable and useful
  - ifreebusy.com, tungle.com, timebridge.com, timetomeet.info, doodle.ch
- Potentially bridge the divide between enterprise calendaring and
  - calendar/scheduling augmenters
  - standalone calendaring clients (no server)

# What we have done

- Standardize/Normalize
- Parameters -URI template
- Error reporting within the HTTP protocol
- Allow for non-authenticated or weakly authenticated service
- Keep it simple (in its simplest form)

# What we have done...

- Outlook compatibility
- Extend?
  - Discovery
  - Authentication
  - Provisioning
  - VAVAILABILITY
    - provide a grouping of available time information over a specific range of time.

# How it works (1)

- The "Read URL" is used to get freebusy data for a user

[http://www.example.com/freebusy/user1@example.com?  
start=20070901T000000-0800](http://www.example.com/freebusy/user1@example.com?start=20070901T000000-0800)

<http://www.example.com/freebusy/user1@example.com>

- returns VFREEBUSY object

# How it works (2)

- The "Publish URL" is used by a client to upload freebusy data for a user

<http://www.example.com/freebusy/user1@example.com>

[http://www.example.com/freebusy?  
user=user1@example.com&token=xcsfdgetdh](http://www.example.com/freebusy?user=user1@example.com&token=xcsfdgetdh)

# What we will show you

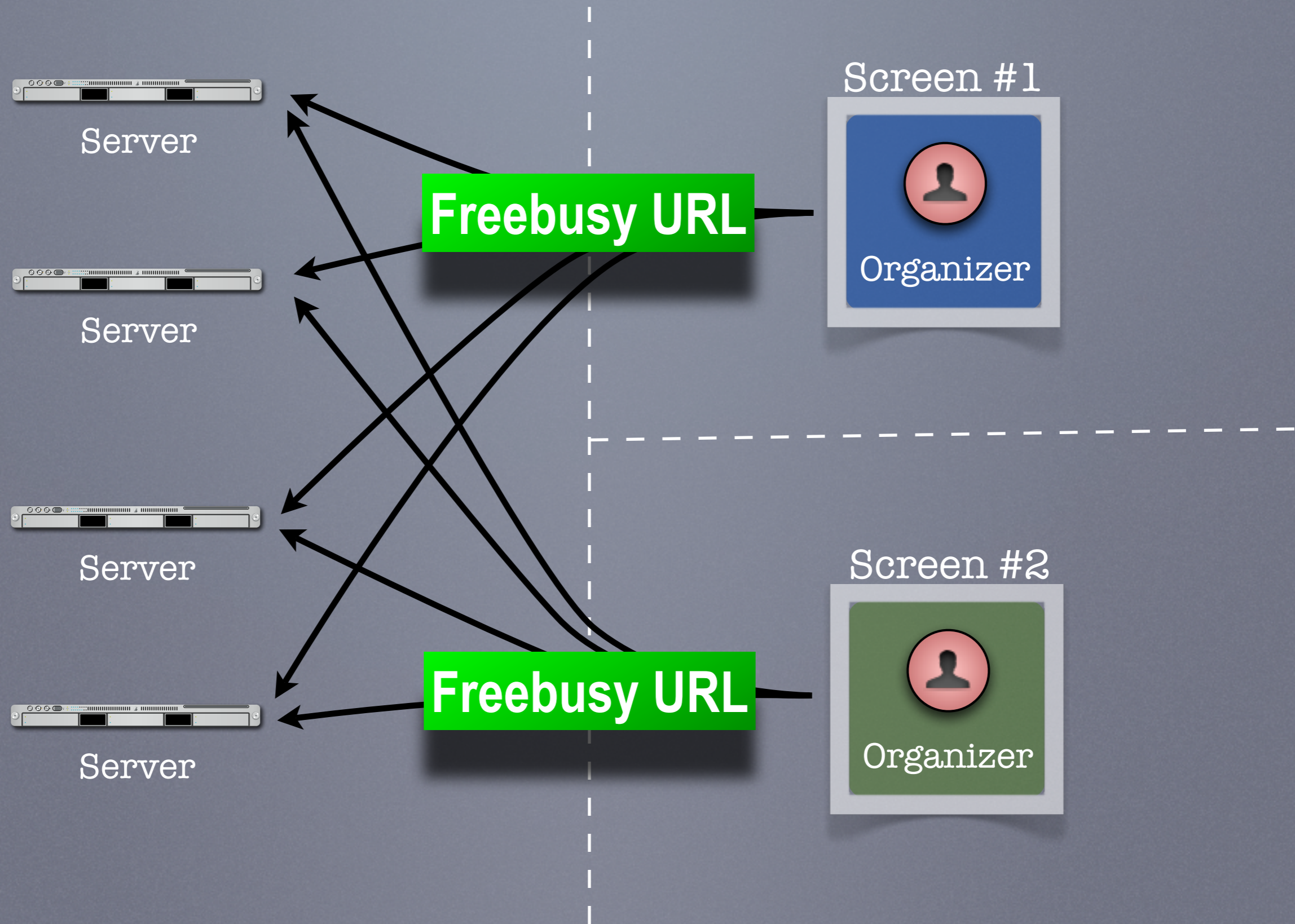
- Basic form FBURL
  - lookups - no publishing
  - Accessing multiple servers from the same clients
  - Comparison with server-server lookups

# Freebusy URL Demonstration #1

Several clients retrieving freebusy  
information



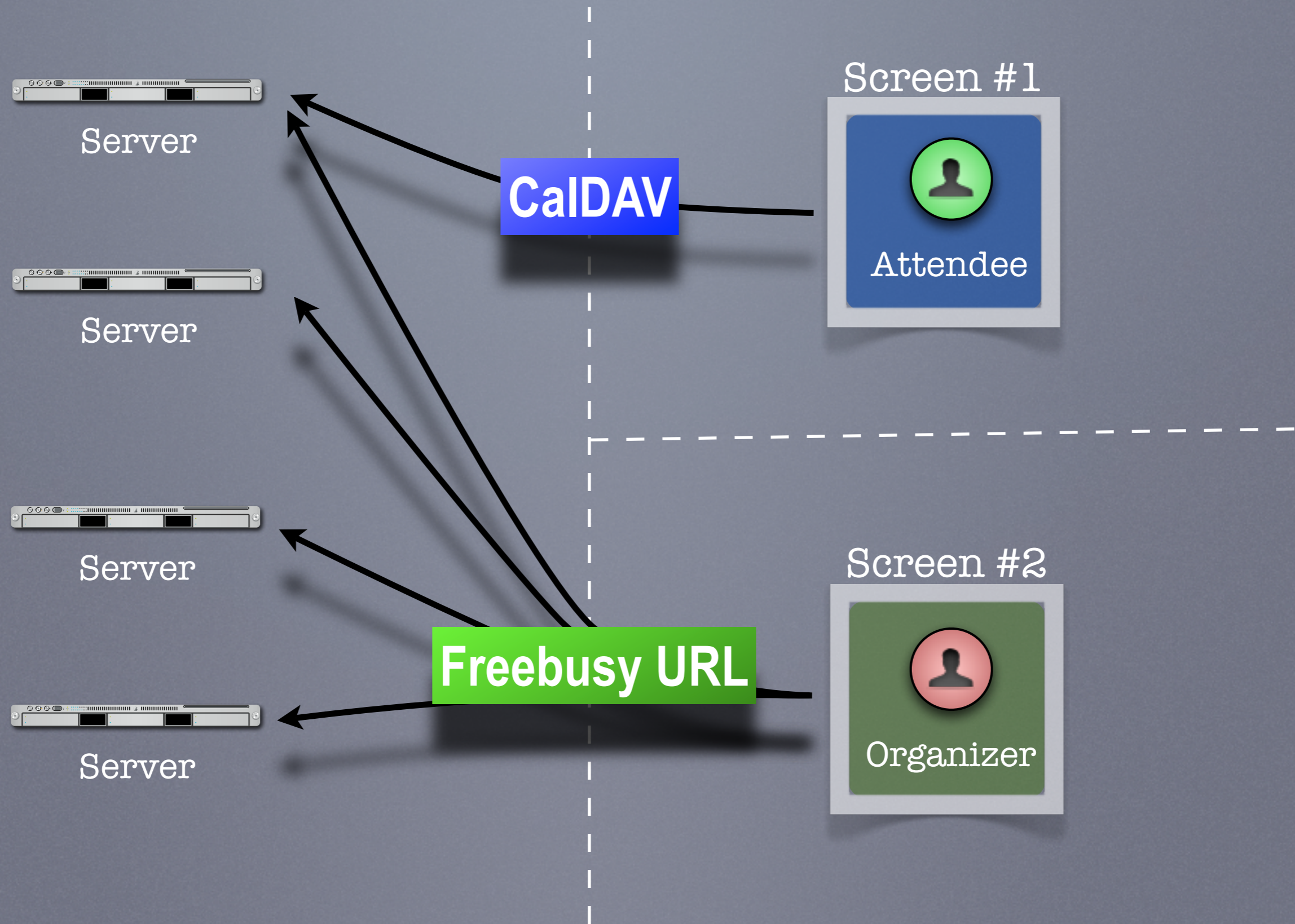
# Freebusy Demo #1 Setup



# Freebusy URL Demonstration #2

Project management aided by freebusy  
information

# Freebusy Demo #2 Setup



# Wrap-up

- We have demonstrated how progress is being made with key scheduling technologies
- As with a lot of CalConnect work this is a very interactive process with specifications and implementations being worked on together
- This ultimately provides for a better specification and interoperability

# CalDAV Scheduling

- Work still needs to be done to fine tune CalDAV scheduling
- Ongoing discussions in TC-CALDAV center around moving most of the scheduling message processing to the server for better reliability
- Hope to complete this by mid-2008

# Realtime Internet Scheduling

- Demonstrated basic scheduling message processing
- Key elements of Realtime Internet Scheduling still need to be developed:
  - Discovery (working on DNS-SD implementation right now)
  - Security - need input from security experts as to what model(s) to use
- Hope to complete this by end of 2008

# Freebusy URL

- Freebusy is LCD scheduling
- Freebusy is soft-core calendaring
- It is what we settle for, not what we want
- But...Free/Busy is very, very useful
- CalConnect will continue to develop FBURL

Thanks