

Evolving the Application Easily, Thanks to DDD



Steve Smith
Force Multiplier for
Dev Teams
[@ardalis](https://twitter.com/ardalis) ardalis.com



Julie Lerman
Software coach,
DDD Champion
[@julielerman](https://twitter.com/julielerman) thedatafarm.com

Overview



- Review current system design**
- Discuss a new feature with customer**
- Introduce message queues**
- Implement the new feature**

Reviewing Our Current System Design

FrontDesk.Blazor x + Front Desk

localhost:5100

Schedule

Clients

Doctors

Rooms

Public Site

Sent Emails

RabbitMQ

Appointments

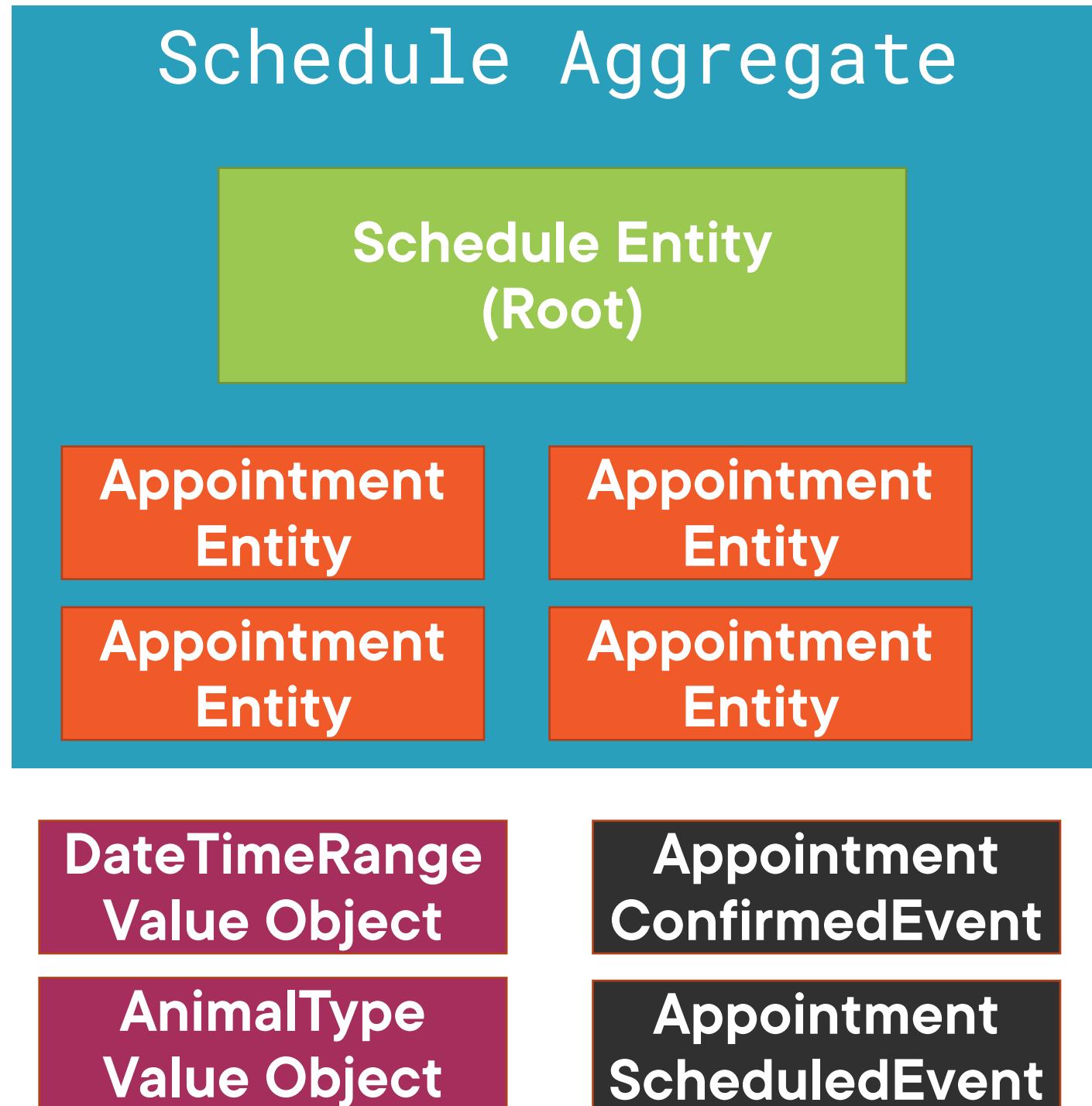
Client: Select a Client...

Today ▶ Monday, September 23, 2030 Day Work Week Agenda

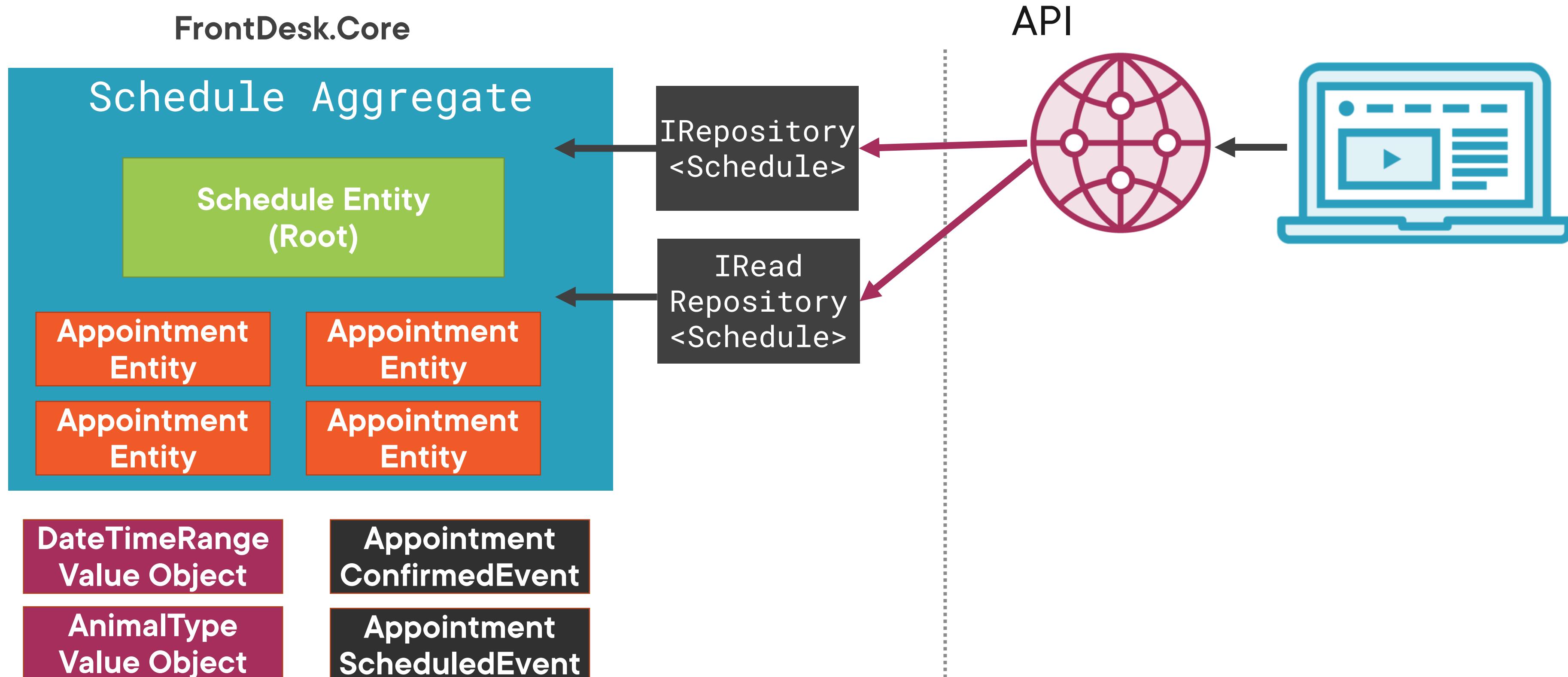
	Exam Room 1	Exam Room 2	Exam Room 3	Exam Room 4	Exam Room 5
	Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23
8:00 AM				(DE) BenFranklin - Michael Jenkins	
9:00 AM				(DE) Sugar - Patrick Neborg	
10:00 AM	(WE) Darwin - Steve Smith (WE) Arya - Steve Smith		(DE) Max - David Batten	(DE) Mim - Shelley Benhoff	
11:00 AM	(WE) Rosie - Steve Smith	(DE) Sampson - Julie Lerman	(DE) Gusto - James Millar	(DE) Jasper - Steve Gordon	

High Level of Front Desk Scheduling App

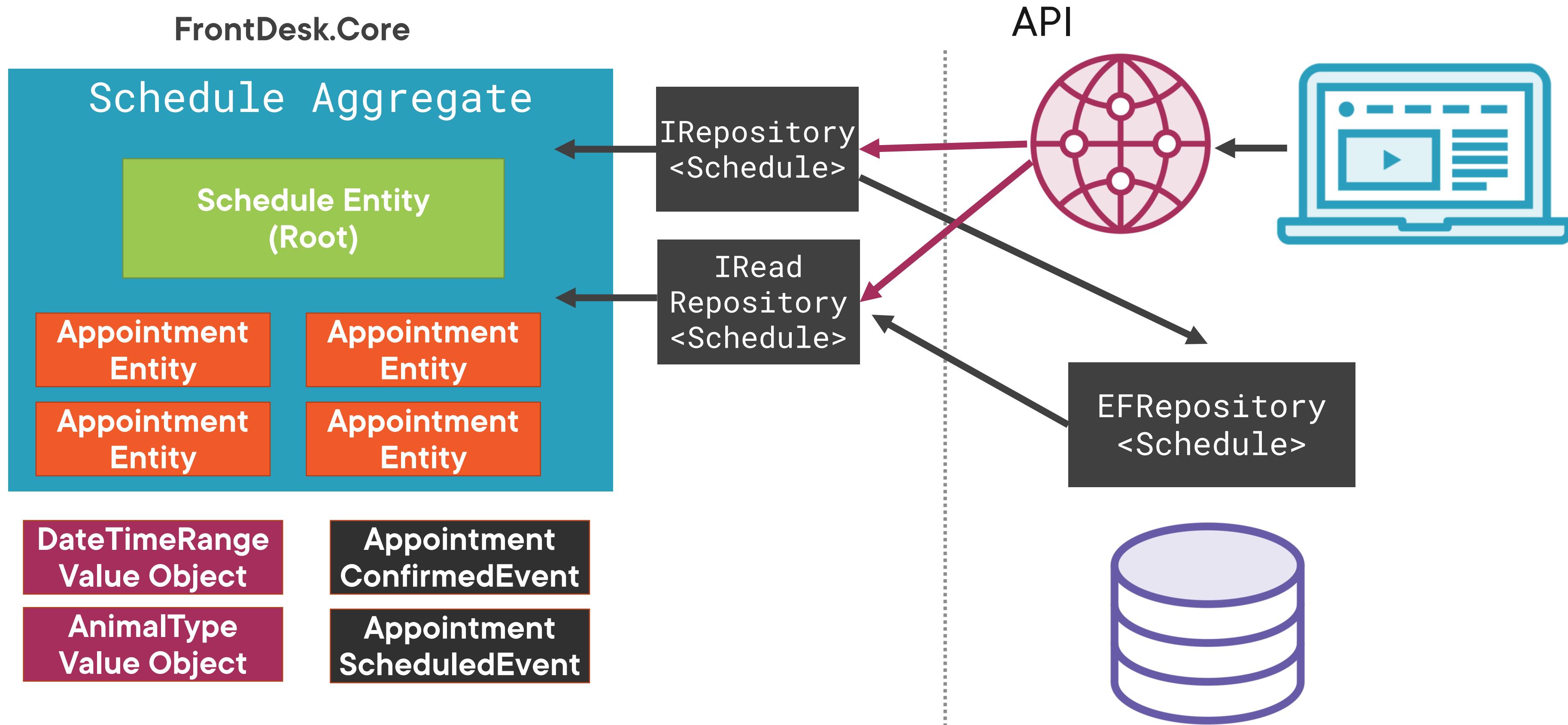
FrontDesk.Core



High Level of Front Desk Scheduling App



High Level of Front Desk Scheduling App



You've learned techniques that will help you expand this application in a clean and uncomplicated way.

Addressing a New Feature with the Domain Expert

Gathering (more) Requirements



Scheduling application is well-received so far

But...clients still frequently forget appointments

Staff doesn't have time to call every client



New Features

Send a confirmation email when the appointment is scheduled

Send a reminder email the day before client's appointment

Provide a link in the emails for the client to confirm the appointment

Planning Our Implementation Steps

Appointment Confirmation

[High Level Workflow]



Using What We've Already Learned & Built

Domain & Integration Events

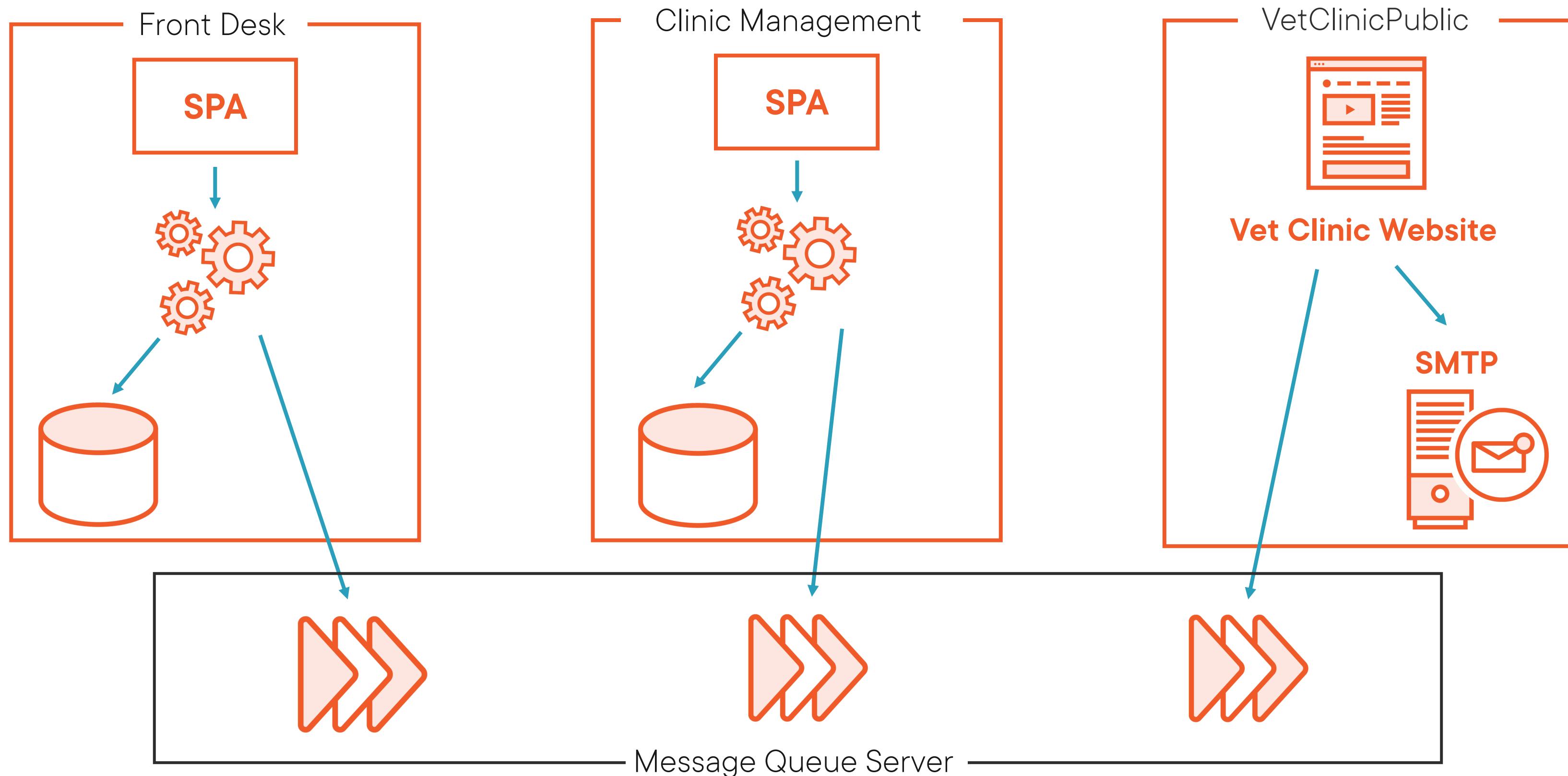
Event Handlers

Services

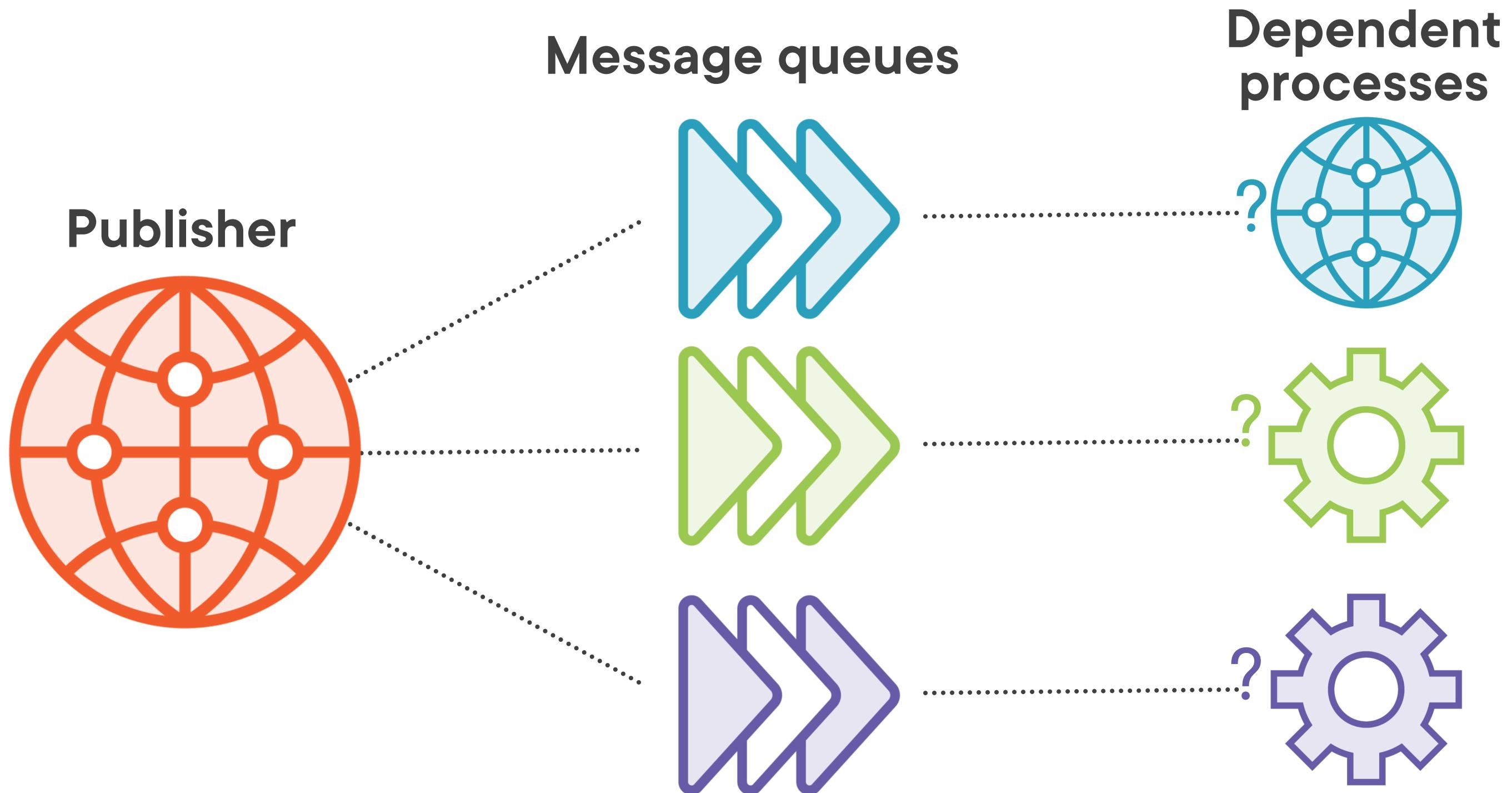
Messaging Queues

Introducing Message Queues

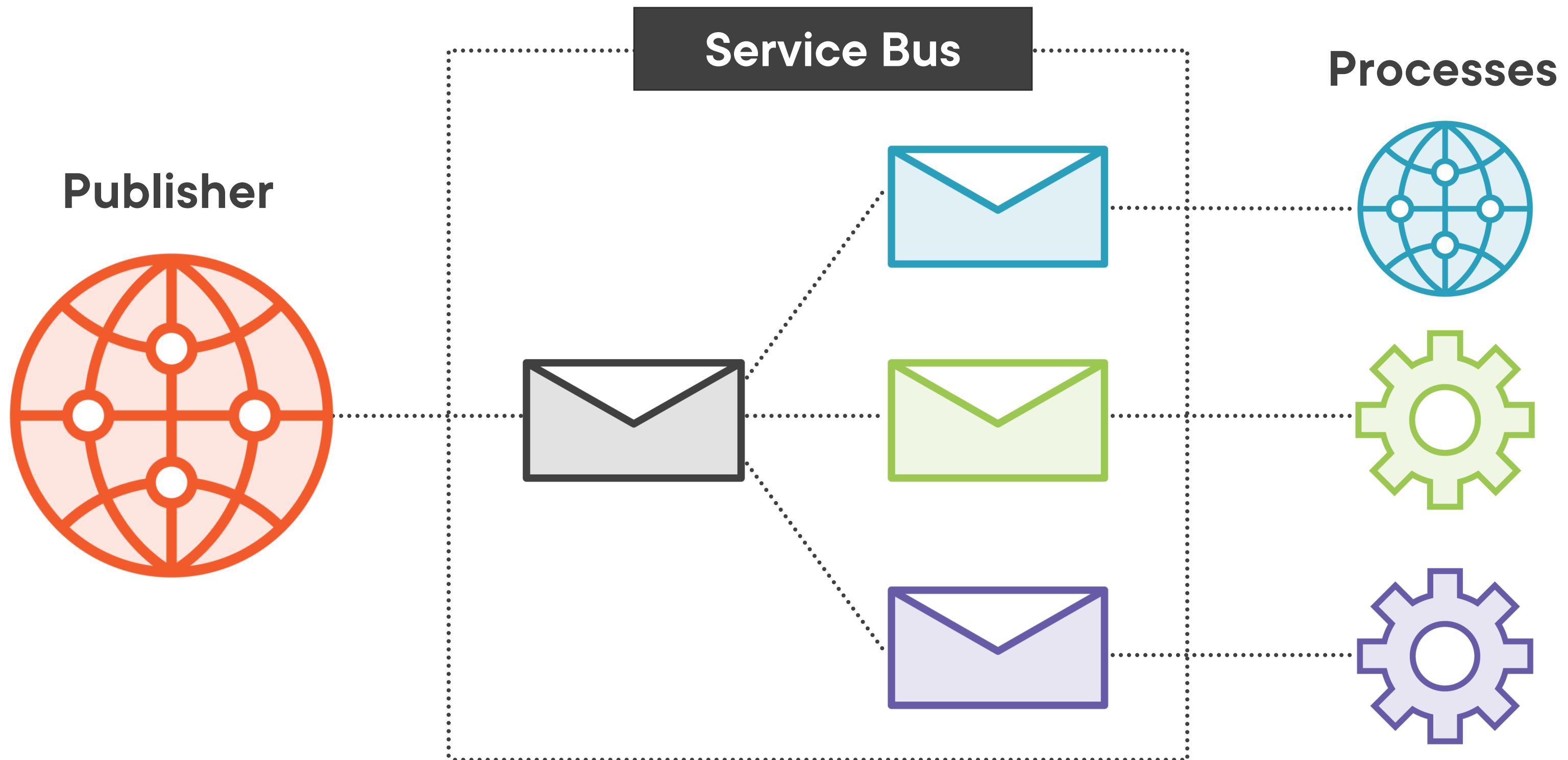
Where Does Message Queue Fit In?



Message Queues



Service Bus



Publish to Service Bus, Regardless of Who May Be Listening

Appointments

Client:

Today

	Exam Room 1	Exam Room 2	Exam Room 3	Exam Room 4
Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23
8:00 AM				(DE) BenFranklin - Michael Jenkins
9:00 AM				(DE) Sugar - Patrick Neborg
10:00 AM	(WE) Darwin - Steve Smith (WE) Arya - Steve Smith		(DE) Max - David Batten (DE) Sampson - Julie Lerman	(DE) Mimic - Steve Smith (DE) Gusto - James Millar (DE) Jasper - Steve Gordon
11:00 AM	(WE) Rosie - Steve Smith	(DE) Sampson - Julie Lerman	(DE) Gusto - James Millar	(DE) Jasper - Steve Gordon

Hello! If anyone out there cares, here's a new appointment that just got scheduled!

Without Service Bus, Create a Specific Message Queue

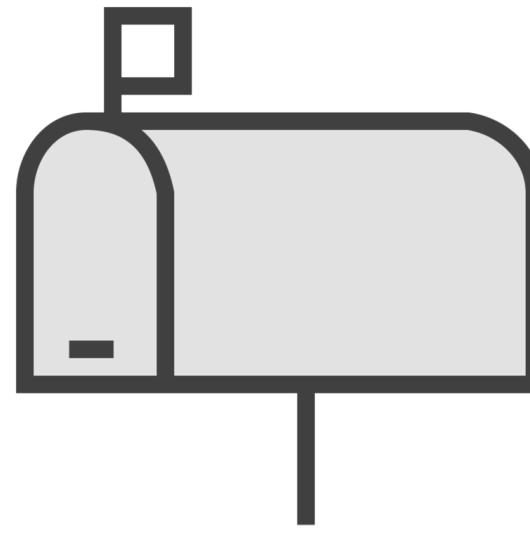
Appointments

Client:

Today

	Exam Room 1	Exam Room 2	Exam Room 3	Exam Room 4
Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23
8:00 AM				(DE) BenFranklin - Michael Jenkins
9:00 AM				(DE) Sugar - Patrick Neborg
10:00 AM	(WE) Darwin - Steve Smith (WE) Arya - Steve Smith		(DE) Max - David Batten (DE) Gusto - James Millar	(DE) Mim - Shelley Benhoff (DE) Jasper - Steve Gordon
11:00 AM	(WE) Rosie - Steve Smith	(DE) Sampson - Julie Lerman		

Hello, Social Media
listener, here's a new
appointment



Rabbit MQ

 RabbitMQ™

Features | Get Started | Support | Community | Docs | Blog



Quorum queues

A webinar on high availability and data safety in messaging

[Learn more](#)

RabbitMQ is the most widely deployed open source message broker.

With tens of thousands of users, RabbitMQ is one of the most popular open source message brokers. From [T-Mobile](#) to [Runtastic](#), RabbitMQ is used worldwide at small startups and large enterprises.

RabbitMQ is lightweight and easy to deploy on premises and in the cloud. It supports multiple messaging protocols. RabbitMQ can be deployed in distributed and federated configurations to meet high-scale, high-availability requirements.

RabbitMQ runs on many operating systems and cloud environments, and provides a [wide range of developer tools for most popular languages](#).

See how other people are using RabbitMQ:

Updates

1. [Erlang 24](#) 24 Mar 2021
[Support and Erlang 22 Retirement Timeline](#)
2. [RabbitMQ 3.8.14](#) 02 Mar 2021
3. [RabbitMQ 3.8.13](#) 27 Feb 2021

[More updates →](#)

Tweets

 RabbitMQ @RabbitMQ

In less than two weeks time, Bintray is going to be shut down. If you provision RabbitMQ from Bintray, please migrate ASAP [blog.rabbitmq.com/posts/2021/03/...](#)

The screenshot shows the Docker Hub interface for the 'rabbitmq' repository. At the top, there's a search bar with the placeholder 'Search for great content (e.g., mysql)'. The navigation bar includes 'Explore', 'Repositories', 'Organizations', 'Get Help', and a user profile for 'julielerman'. Below the header, the breadcrumb navigation shows 'Explore > rabbitmq'. The main content area features the 'rabbitmq' logo and a star icon, followed by the text 'Docker Official Images'. A brief description states 'RabbitMQ is an open source multi-protocol messaging broker.' To the left, there's a download count of '1B+' and a list of supported platforms: Container, Linux, IBM Z, ARM 64, ARM, x86-64, PowerPC 64 LE, 386, Messaging Services, and Official Image. On the right, there's a section titled 'Copy and paste to pull this image' containing the command 'docker pull rabbitmq' with a copy icon, and a link to 'View Available Tags'. Below this, there are tabs for 'Description' (which is selected), 'Reviews', and 'Tags'. A large 'Quick reference' section follows, containing two bullet points: 'Maintained by: the Docker Community' and 'Where to get help: the Docker Community Forums, the Docker Community Slack, or Stack Overflow'. Another section, 'Supported tags and respective Dockerfile links', lists three tags: '3.8.15-rc.2', '3.8-rc', '3.8.15-rc.2-management', '3.8-rc-management', and '3.8.15-rc.2-alpine', each with a corresponding Dockerfile link.

rabbitmq ☆

RabbitMQ Messaging that just works

Docker Official Images

RabbitMQ is an open source multi-protocol messaging broker.

↓ 1B+

Container Linux IBM Z ARM 64 ARM x86-64 PowerPC 64 LE 386 Messaging Services Official Image

Description Reviews Tags

Quick reference

- Maintained by: the Docker Community
- Where to get help: the Docker Community Forums, the Docker Community Slack, or Stack Overflow

Supported tags and respective Dockerfile links

- 3.8.15-rc.2 , 3.8-rc
- 3.8.15-rc.2-management , 3.8-rc-management
- 3.8.15-rc.2-alpine , 3.8-rc-alpine

No need to
install, just use
a Docker
container!

Overview

▼ Totals

Queued messages last minute ?

Currently idle

Message rates last minute ?

Currently idle

Global counts ?

Connections: 0

Channels: 0

Exchanges: 7

Queues: 0

Consumers: 0

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
a@localhost	34 467109 available	0 420306 available	409 1048576 available	92MB 6.1GB high watermark	13GB 48MB low watermark	8m 57s	basic disc 6 rss	This node All nodes	
b@localhost	34 467109 available	0 420306 available	409 1048576 available	89MB 6.1GB high watermark	13GB 48MB low watermark	14m 28s	basic disc 6 rss	This node All nodes	

► Churn statistics

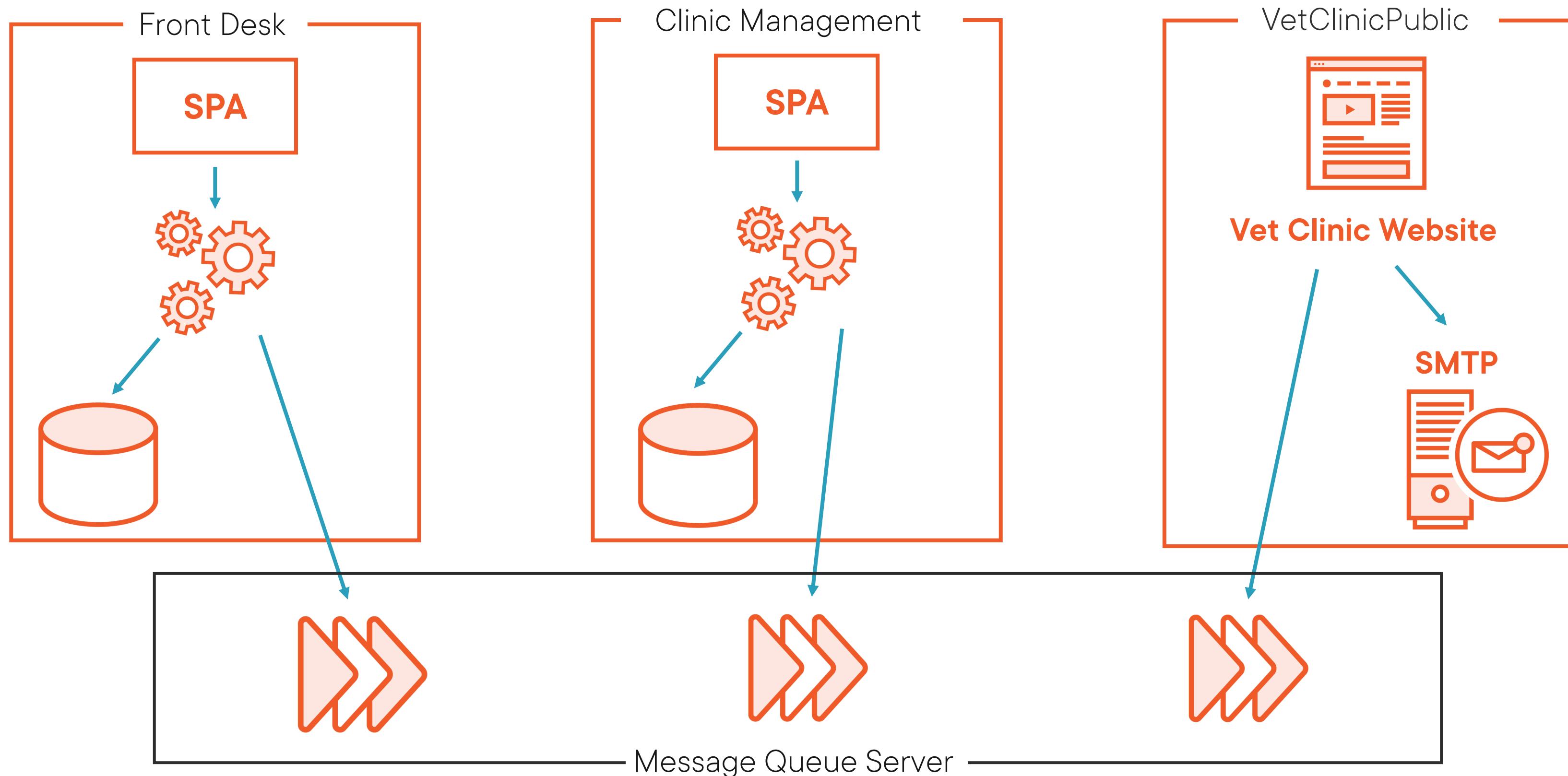
► Ports and contexts

► Export definitions

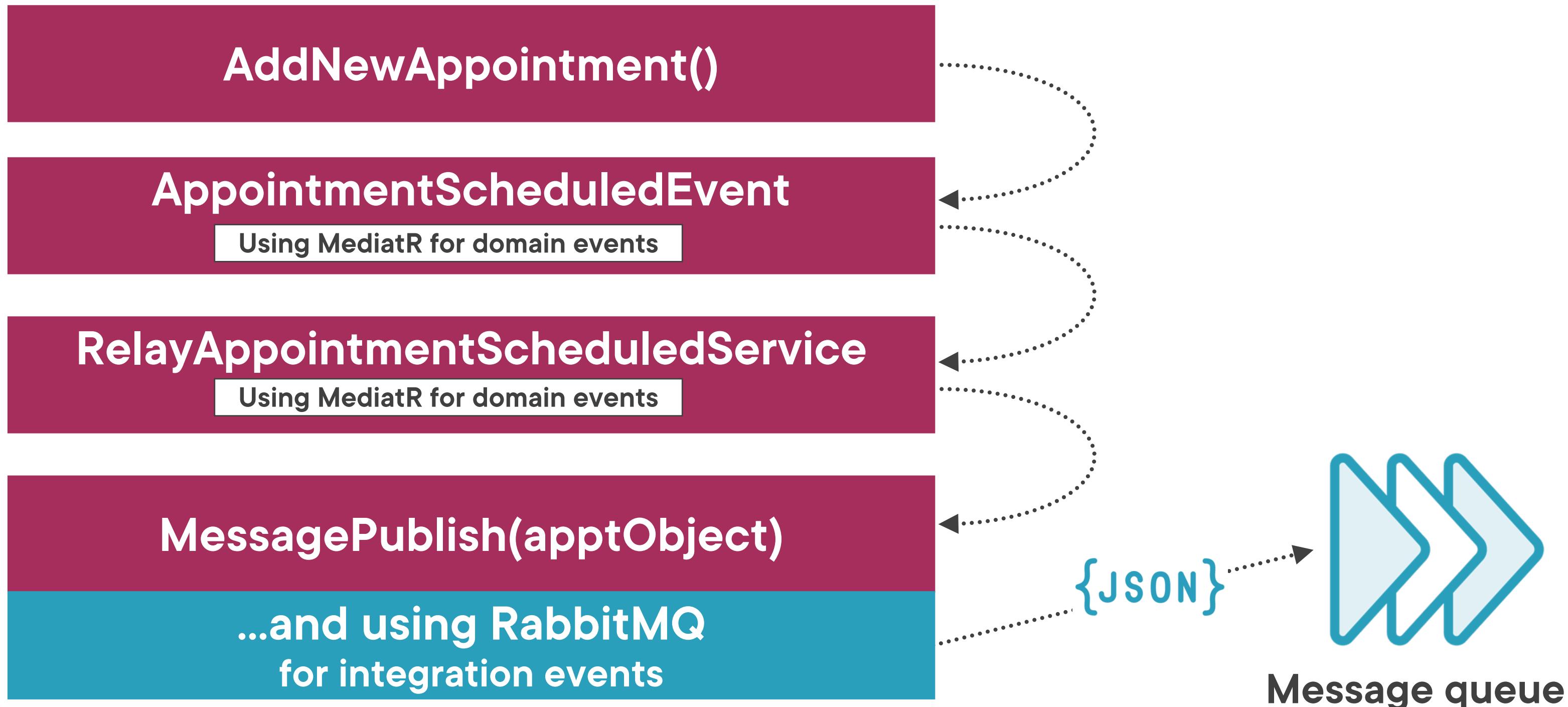
► Import definitions

Sending a Message to the Queue

Where Does a Message Queue Fit In?

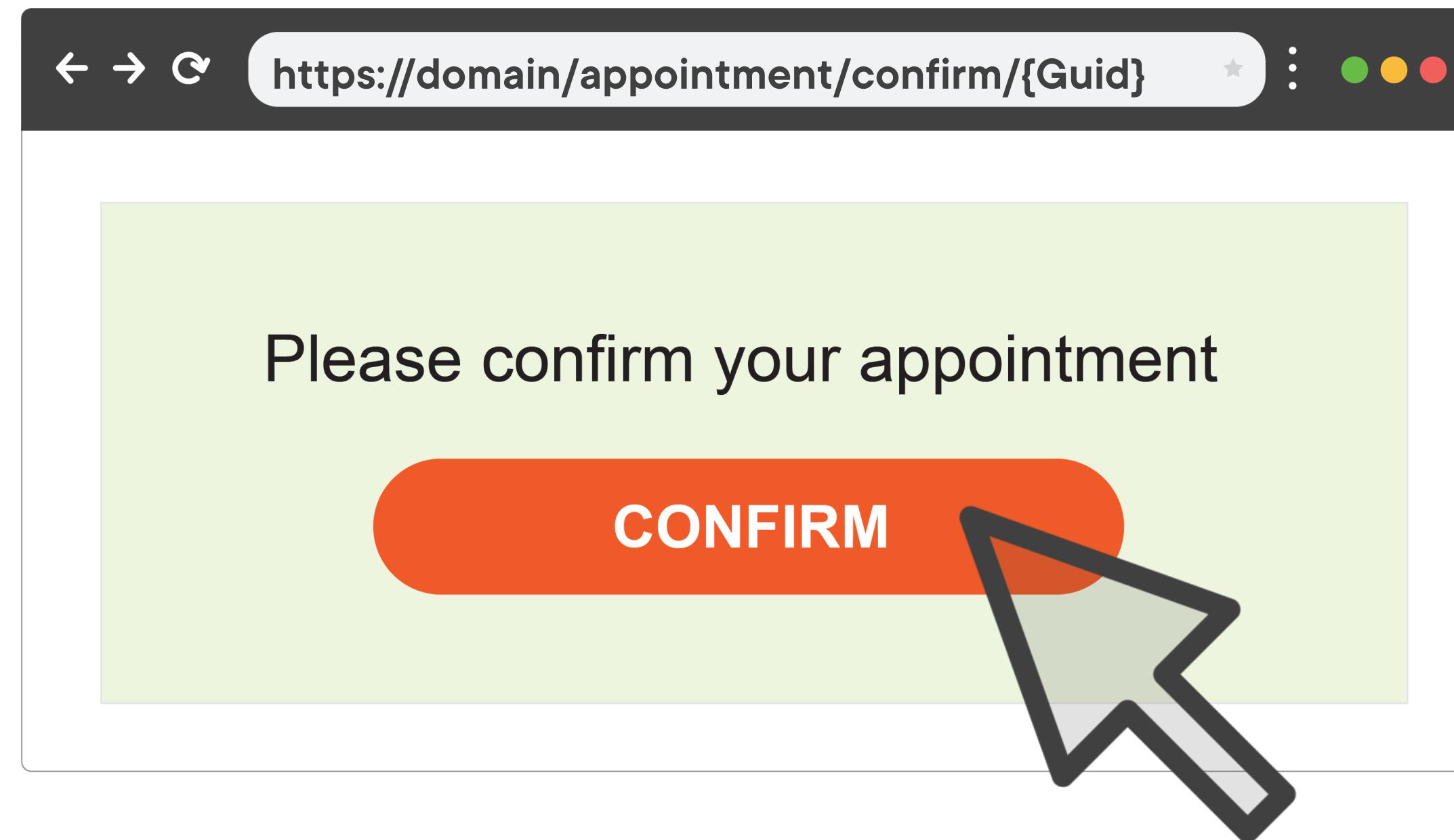


Appointment Scheduled Workflow



Reading From the Message Queue and Acting on the Message

Email and Confirmation Logic on Public Website



Send Email via Message Queue



For messages found, retrieves appointment details and...



Send confirmation email

ConfirmationEmailSender.cs

```
public void SendConfirmationEmail(Models.SendAppointmentConfirmationCommand appointment)
{
    _logger.LogInformation($"Sending email to confirm appointment: {appointment}");
    string confirmUrl =
        $"http://localhost:{_config.Port}/appointment/confirm/{appointment.AppointmentId}";
    string to = appointment.ClientEmailAddress;
    string from = "donotreply@thevetclinic.com";
    string subject = "Vet Appointment Confirmation for " + appointment.PatientName;
    string body = String.Format("<html><body>Dear {0},<br/><p>Please click the link below
to confirm {1}'s appointment for a {2} with {3} on {4}.</p><p>Thanks!</p><p><a
href='{5}'>CONFIRM</a></p><p>Please call the office to reschedule if you will be unable to
make it for your appointment.</p><p>Have a great day!</p></body></html>",
        appointment.ClientName, appointment.PatientName, appointment.AppointmentType,
        appointment.DoctorName, appointment.AppointmentStartTime.ToString(), confirmUrl);
    _emailSender.SendEmail(to, from, subject, body);
}
```

Papercut-SMTP

Can be used to
emulate email
sending

Free & open
source

Papercut-SMTP | Papercut SMTP x +

← → ⌛ 🔒 papercut-smtp.com

View on GitHub

Papercut-SMTP

Papercut SMTP—The Simple Desktop Email Server

Papercut  The Simple Desktop Email Helper

[build](#) passing [gitter](#) [join chat](#) [Say Thanks!](#)

The problem

If you ever send emails from an application or web site during development, you're familiar with the fear of an email being released into the wild. Are you positive none of the 'test' emails are addressed to colleagues or worse, customers? Of course, you can set up and maintain a test email server for development – but that's a chore. Plus, the delay when you are waiting to view new test emails can radically slow your development cycle.

Papercut to the rescue!

Papercut is a 2-in-1 quick email viewer AND built-in SMTP server (designed to receive messages only). Not only does it not enforce any restrictions how you prepare your email, but it allows you to view the whole email-chilada: body, html, headers, attachment down right down to the naughty raw encoded bits. Papercut can be configured to run on startup and sit quietly (minimized in the tray) only providing a notification when a new message has arrived.

Download Now

[Latest Release](#)

Download Papercut.Setup.exe installer.

Using Multiple Queues to Handle Various Communications

Multiple Queues in Our System

Front Desk

Appointments

Client: Select a Client...

Today | Monday, September 23, 2030

	Exam Room 1	Exam Room 2	Exam Room 3	Exam Room 4
Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23	Mon 9/23
8:00 AM				
9:00 AM				
10:00 AM	(WE) Darwin - Steve Smith (WE) Arya - Steve Smith (WE) Rosie - Steve Smith	(DE) Sampson - Julie Lerman	(DE) Max - David Batten (DE) Gusto - James Millar	(DE) BenFranklin - Michael Jenkins (DE) Sugar - Patrick Neborg (DE) Min - Shelley Benhoff (DE) Jasper - Steve Gordon
11:00 AM				



Vet Clinic Public

localhost:7100/appointment/confirm/da34ffc6-1f7b-4503-be9c-5933e4da3d5b

Home Home

Your Vet Clinic

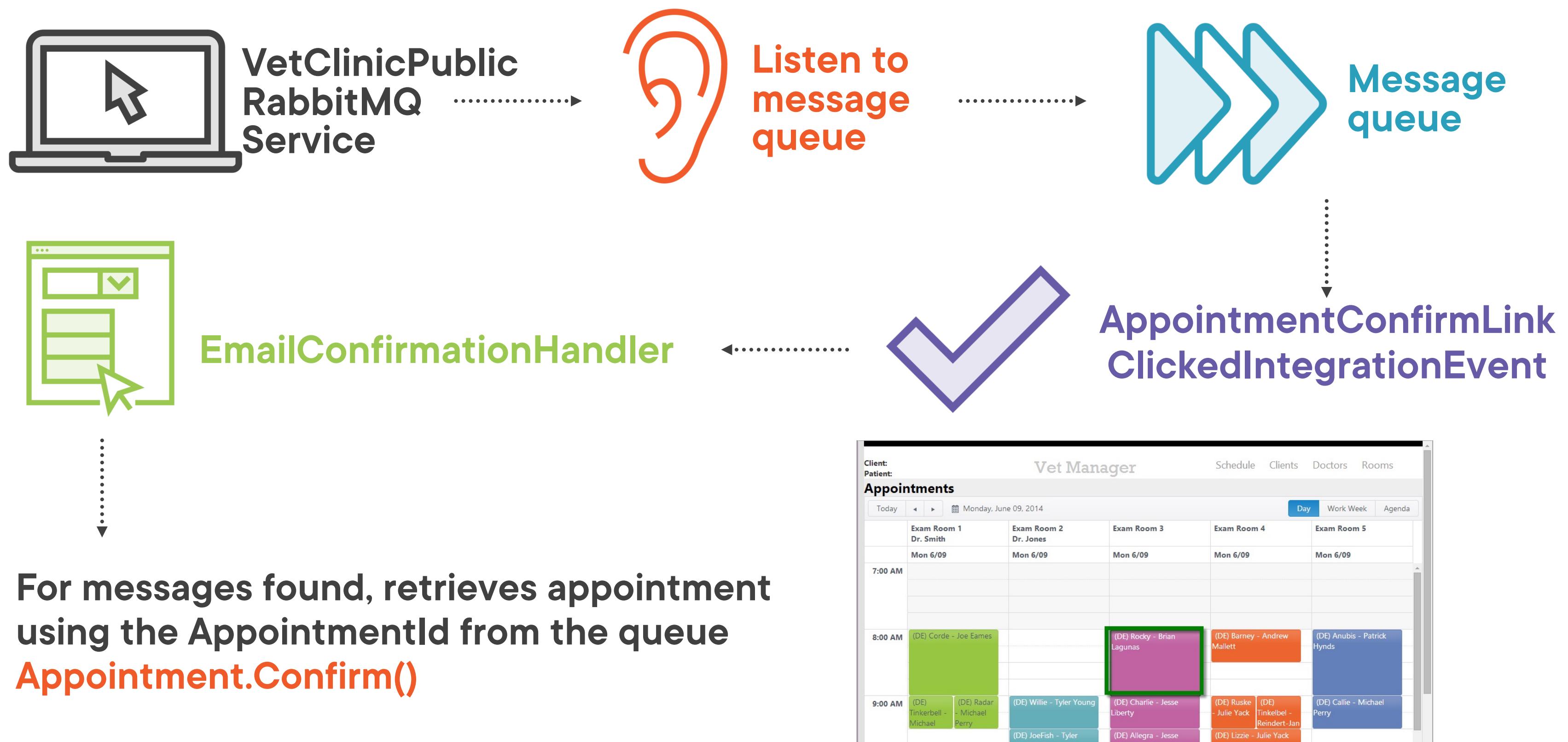
Taking great care of your pets

Appointment Confirmed!

Thanks! We'll see you and your pet soon!

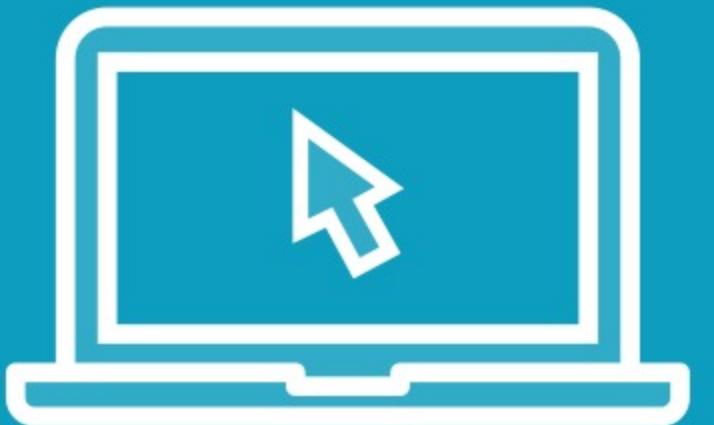
© 2021 - Your Vet Clinic

Sending the Confirmation to Front Desk



Debugging to See the Detailed Implementation in Code

Demo



Considering Microservices

Microservices Basic Tenets

Self-contained

No dependency on other microservices

Independently deployable

Changing internals should not break communications

Microservice design can learn
a lot from
Domain-Driven Design



Bounded context
is not necessarily equal to a
microservice



Bounded Contexts, Microservices and Everything In Between

Vladik Khononov at KanDDDinsky 2018

bit.ly/vladonmicro (YouTube)

Microservices Basic Tenets

FrontDeskRabbitMQService



Self-contained



No dependency on other microservices



Independently deployable



Changing internals should not break communications

Sharing Some Tips for Extending and Running the Sample Application

Extra Credit – Reminder Emails Service



Runs once each morning



Sends out text messages for appointments scheduled for the next day



Avoid sending duplicate emails if run multiple times



Avoid sending to clients who haven't opted in

Considering the UI in the Domain Design



Is it an anti-pattern to think about the UI when
we are focused on the domain?

No!

The demands of the user interface can impact parts of your application.



Jimmy Nilsson

Factor10.com

Author of “Applying
Domain-Driven Design
and Patterns”

“Most likely I’m learning stuff [with UI sketching] that affects my story and scenario”

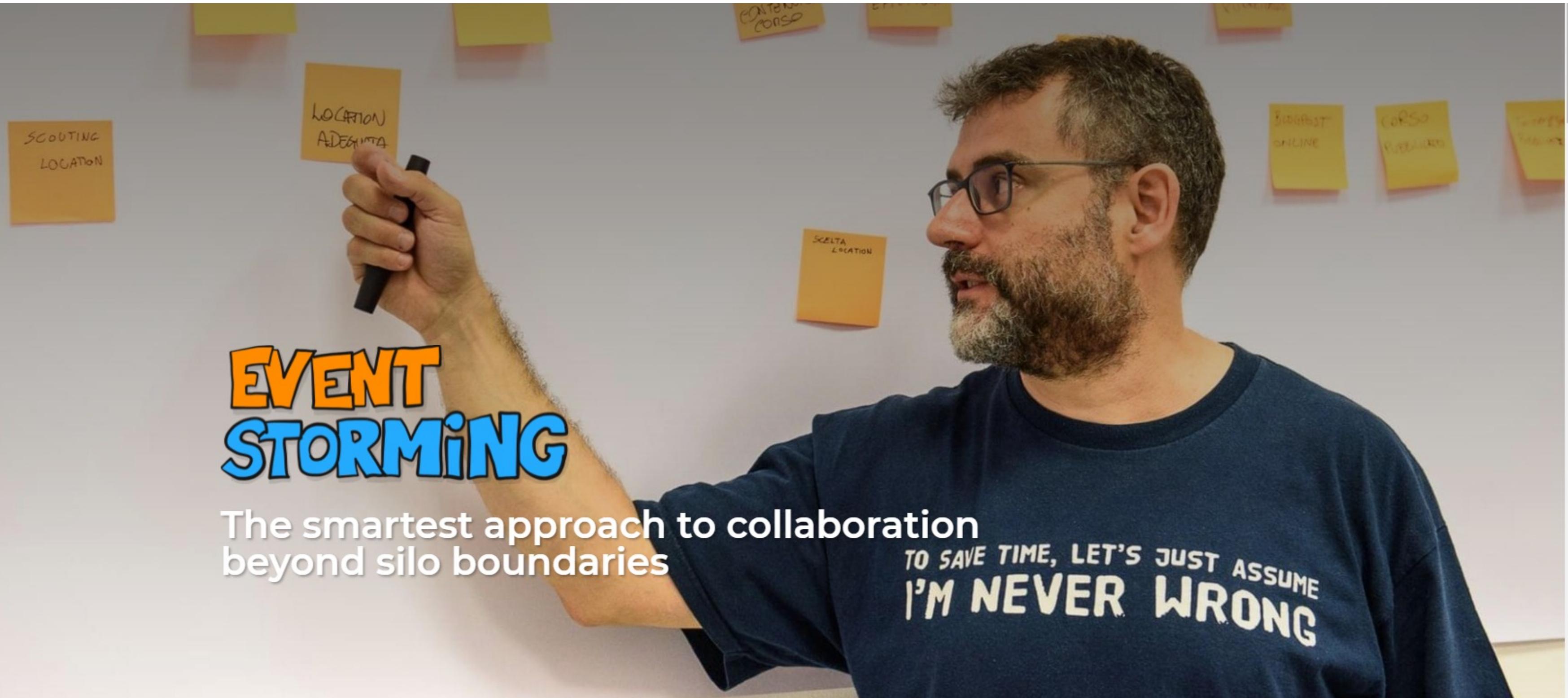
Jimmy Nilsson

Developing Core Business Applications with DDD and Microsoft .NET
TechEd NA 2013

Scheduling is not our domain.

Modeling with Event Storming and Other Techniques

Event Storming



Alberto Brandolini

eventstorming.org

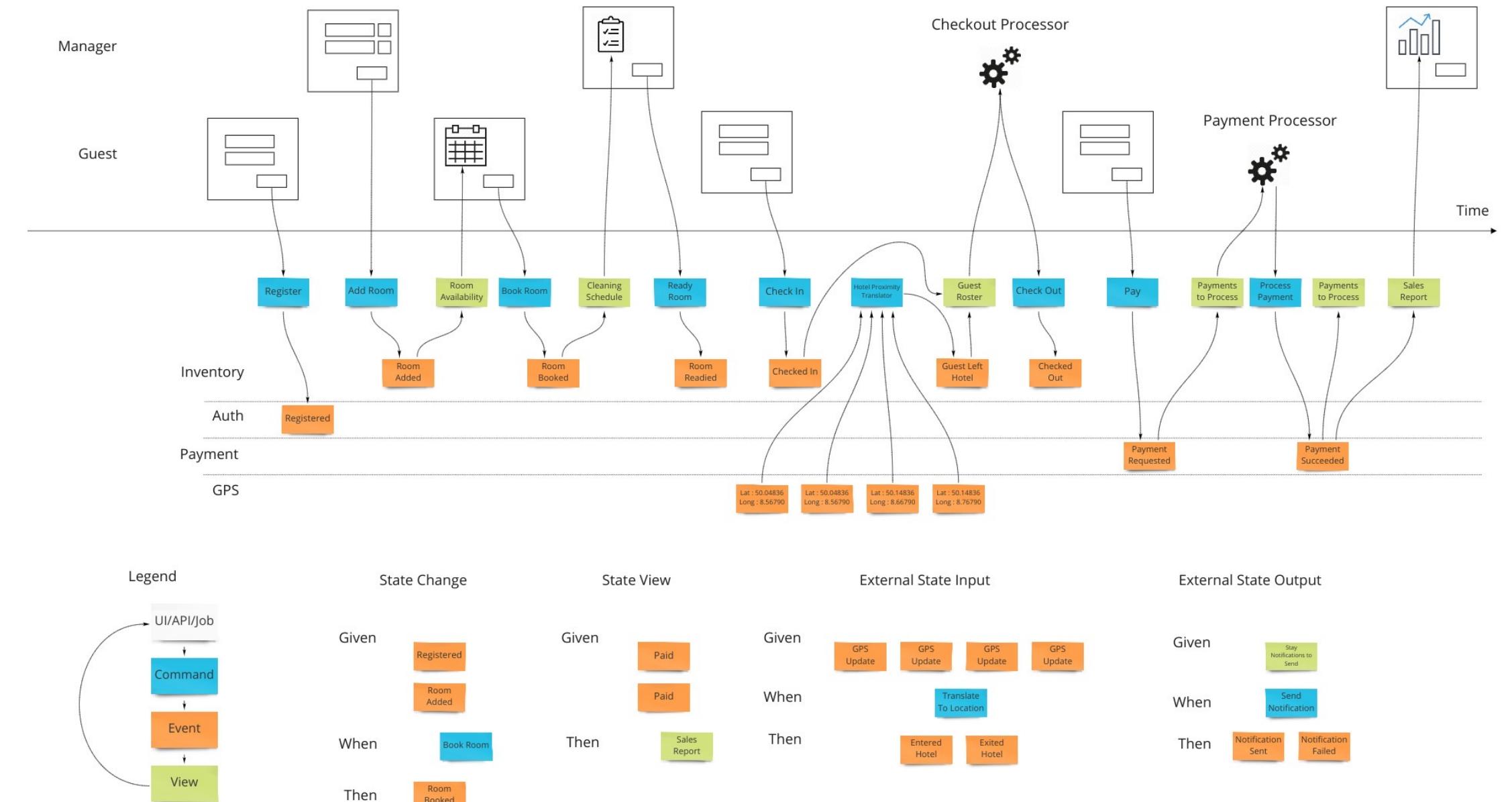
Chaotic Discovering During Event Storming



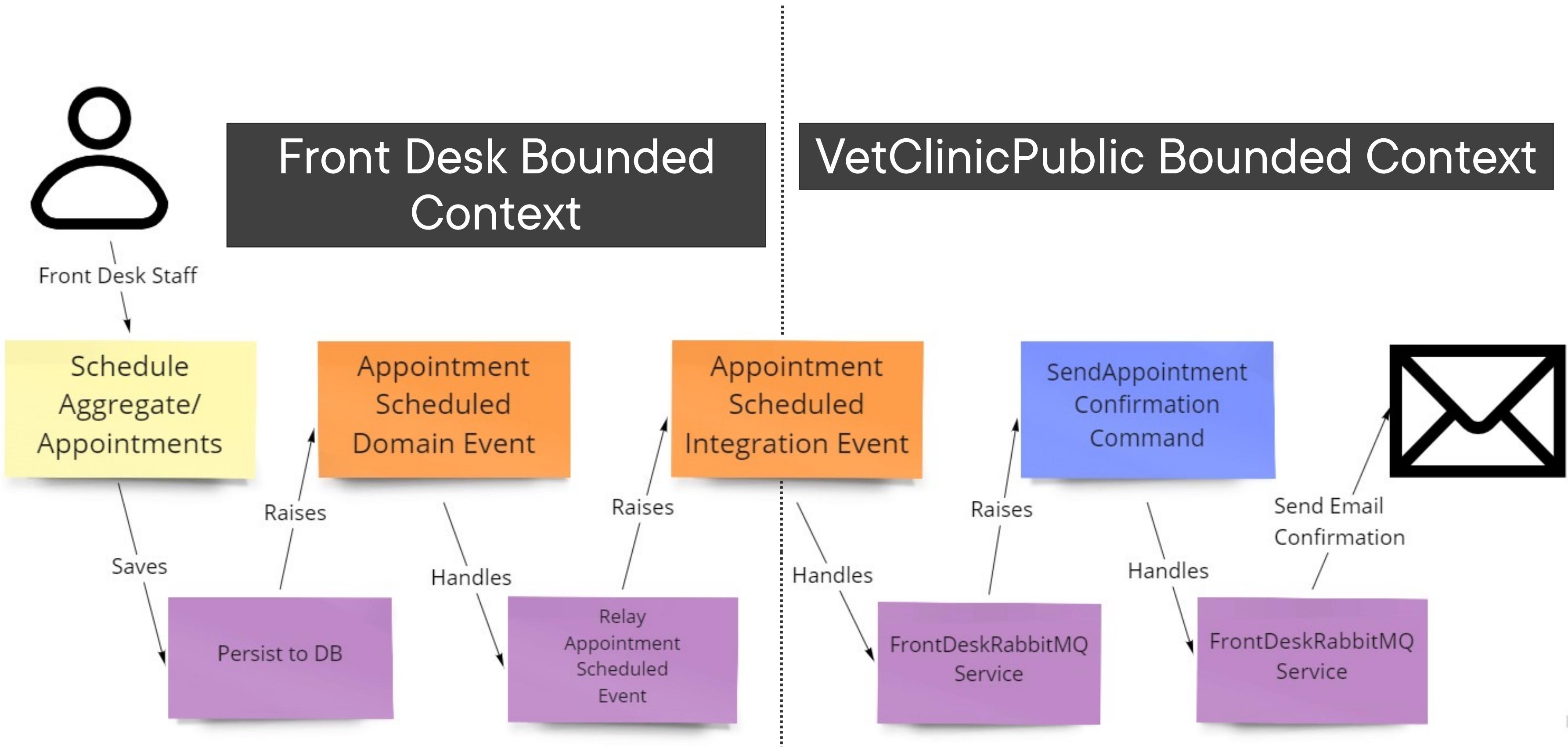
Event Modeling



Adam Dymitruk
EventModeling.org



A Model Designed Using Miro.com



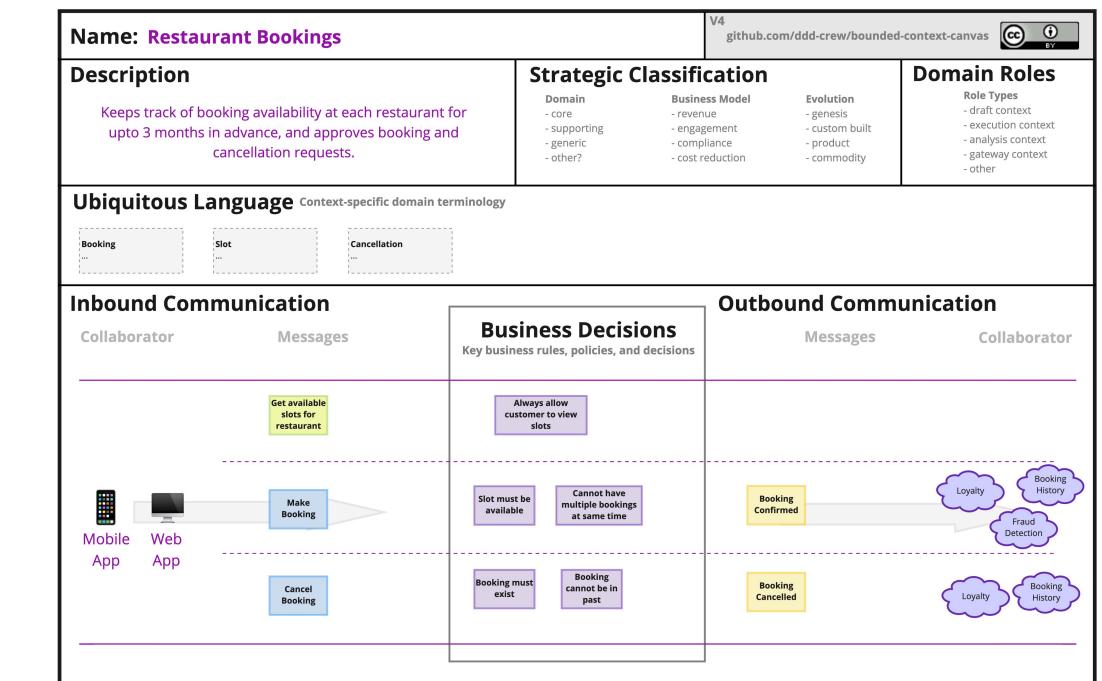
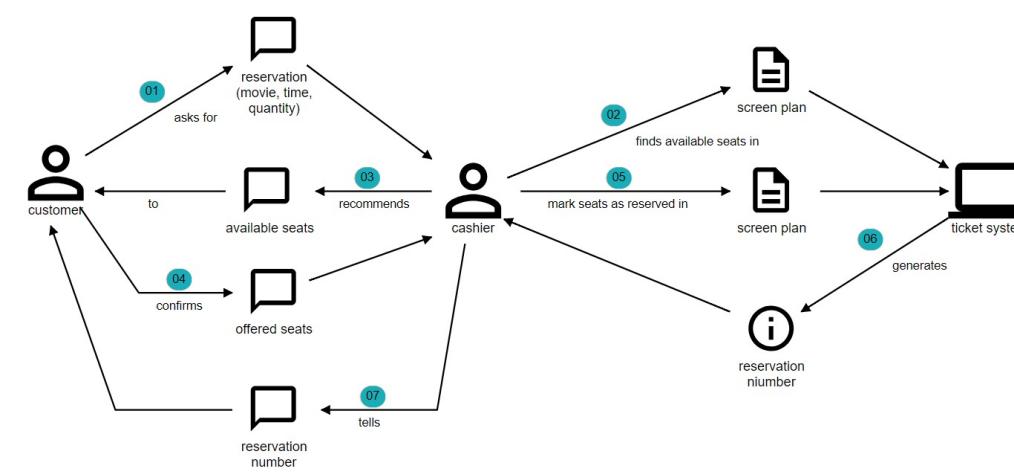
Some More Modeling Processes & Tools



The best way to learn a language is to listen to other people speak that language. Try to repeat what you hear and mind their feedback. Gradually, you will progress from individual words to phrases and to complete sentences. The more you speak, the faster you will learn.

With **Domain Storytelling** you can employ the same principle when learning a new domain language. Let domain experts tell their domain stories. While listening, you record the domain stories using a pictographic language. The domain experts can see immediately if you understand their story correctly. After very few stories, you are able to talk about the people, tasks, tools, work items, and events in that domain.

The best way to explain Domain Storytelling is to see it in action.
Watch this live modeling session (hosted by the Virtual Domain-Driven Design Meetup):



Domain Storytelling

domainstorytelling.org

Domain Storytelling Modeler

wps.de/modeler

Bounded Context Canvas

github.com/ddd-crew/bounded-context-canvas



Beware!

Analysis Paralysis

Eric Evans on the Fallacy of Perfectionism

Domain-Driven Design
provides guidance, not rules.

“There’s something about DDD that brings out the perfectionist in people... they say, this model’s not really good enough...and basically churn and churn. I’m here to say, no model is going to be perfect.”

Eric Evans

“We need to know what we’re doing with this thing, the scenarios we’re trying to address. We want a model that helps us do that, that makes it easier to make software that solves those problems. That’s it.”

Eric Evans

All models are wrong.

Some models are useful.

Lessons Learned Since Our 2014 Course

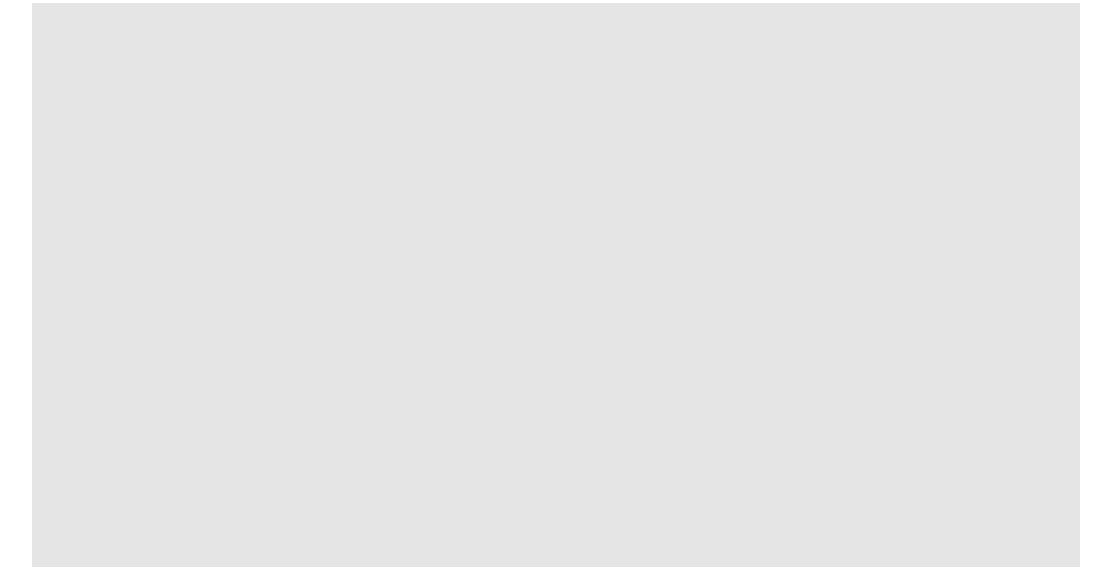
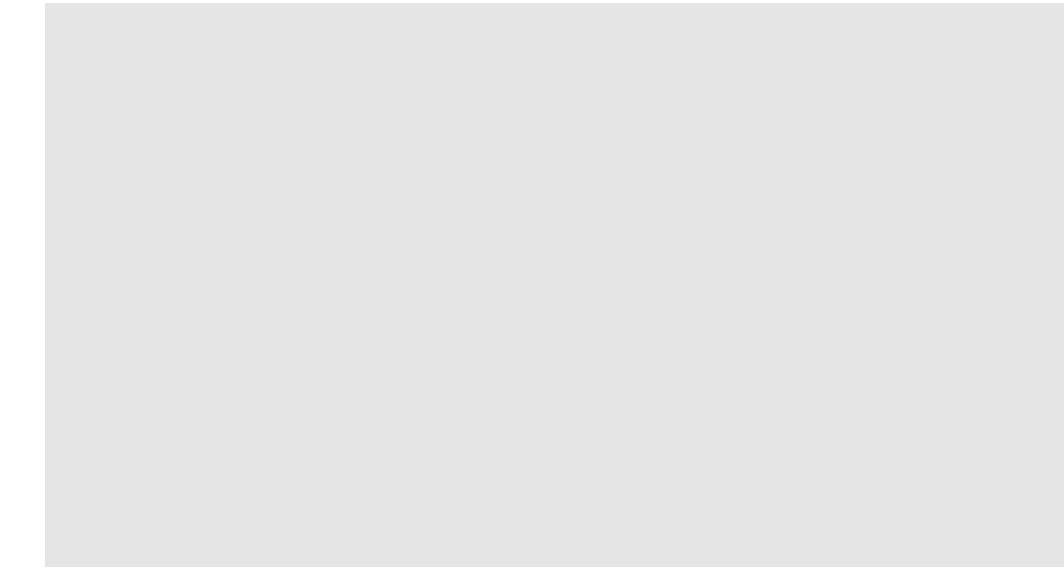
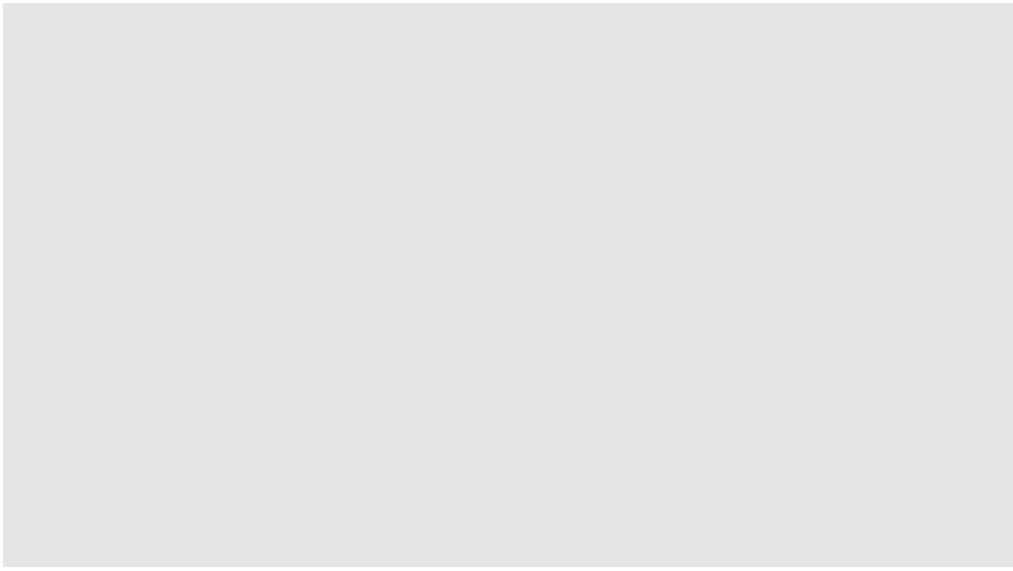
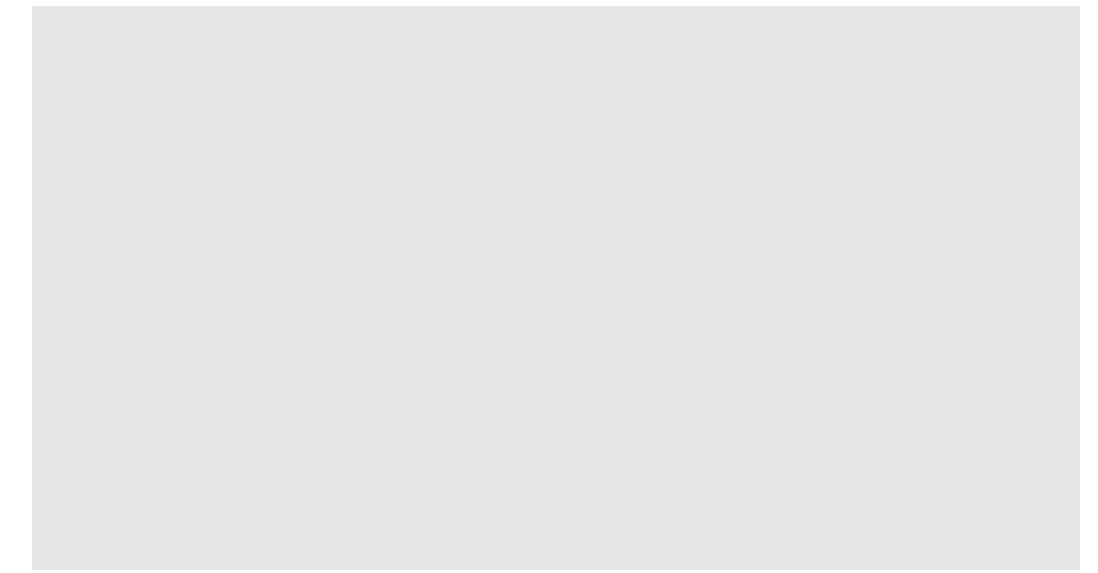
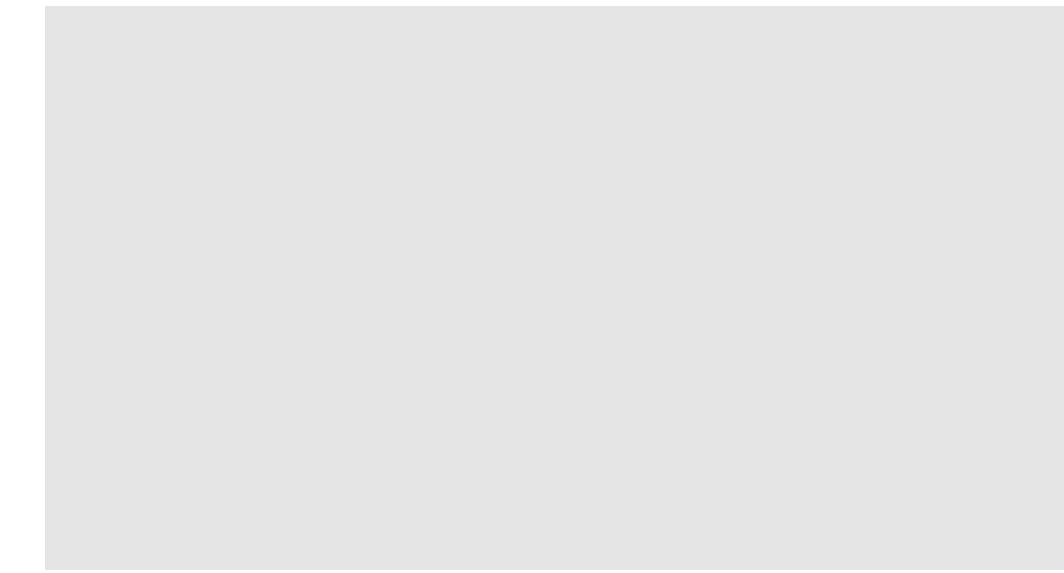
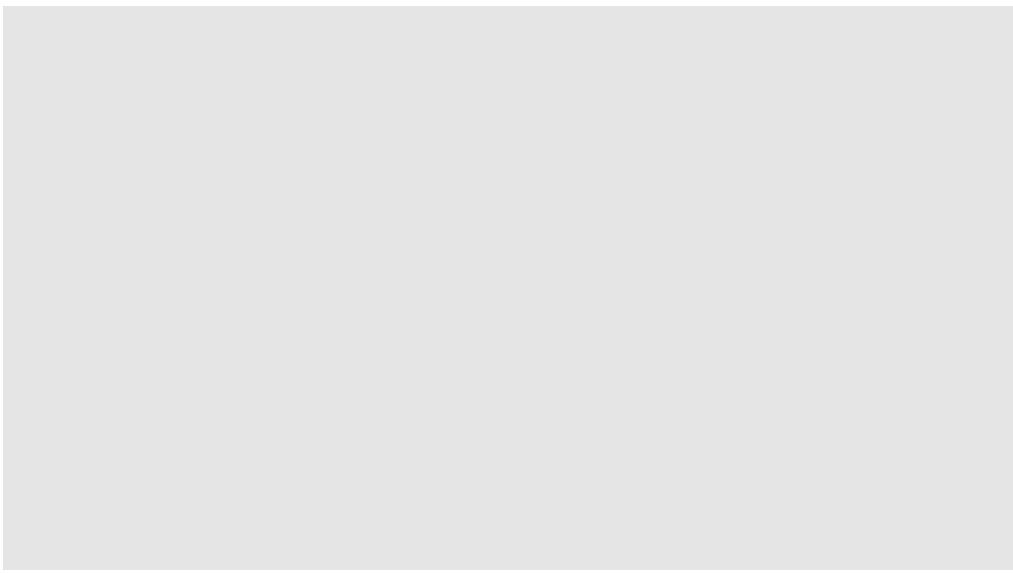
Domain-Driven Design Fundamentals

(as seen when originally published in 2014)

The screenshot shows the Pluralsight website interface. At the top, there's a navigation bar with links for INDIVIDUALS, BUSINESS, ACADEMIC, FREE TRIAL, BLOG, SUPPORT, and Sign in. Below the navigation is a search bar labeled "Search the library". The main content area features a course thumbnail for "Domain-Driven Design Fundamentals", which includes a clock icon and the category "Software Practices". To the right of the thumbnail, the course title is displayed in bold red text. A brief description follows, mentioning that the course teaches the fundamentals of Domain-Driven Design (DDD) through customer interactions and a demo application, with advice from Eric Evans. On the far right, course metadata is listed: Authored by Smith, Lerman; Duration: 4h 16m; Level: Intermediate; Released: 6/25/2014; and Course Rating: ★★★★★. Below the course details are social sharing buttons for Twitter and LinkedIn. At the bottom of the page, a table lists the course's contents with columns for Progress and Duration.

	Progress	Duration
expand all collapse all		
▶ Introducing DDD		00:24:18
Introduction		00:23
10 Years of Domain-Driven Design		01:00
What's in This Course?		01:27
Eric Evans' Participation in This Course		00:29
Why Should You Care About Domain-Driven Design?		02:13

Thanks!



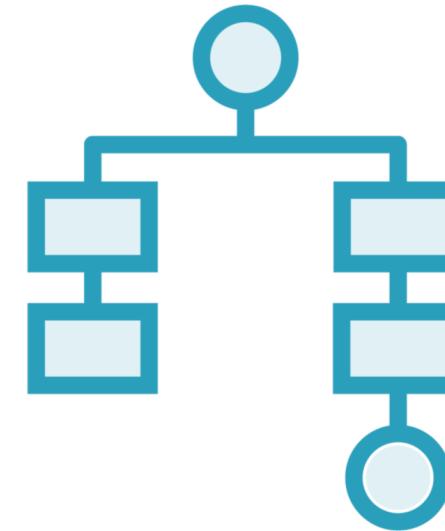
fds

DDD Is Growing and Evolving

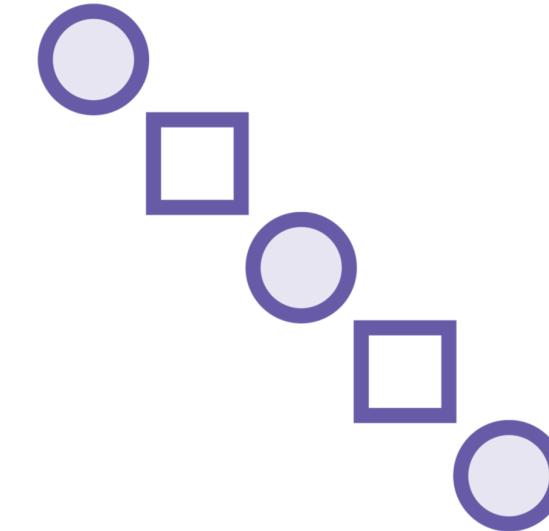
New practices and patterns are gaining popularity as teams find success using them



Event Storming



Event Modeling



Patterns

New Technologies

.NET 5

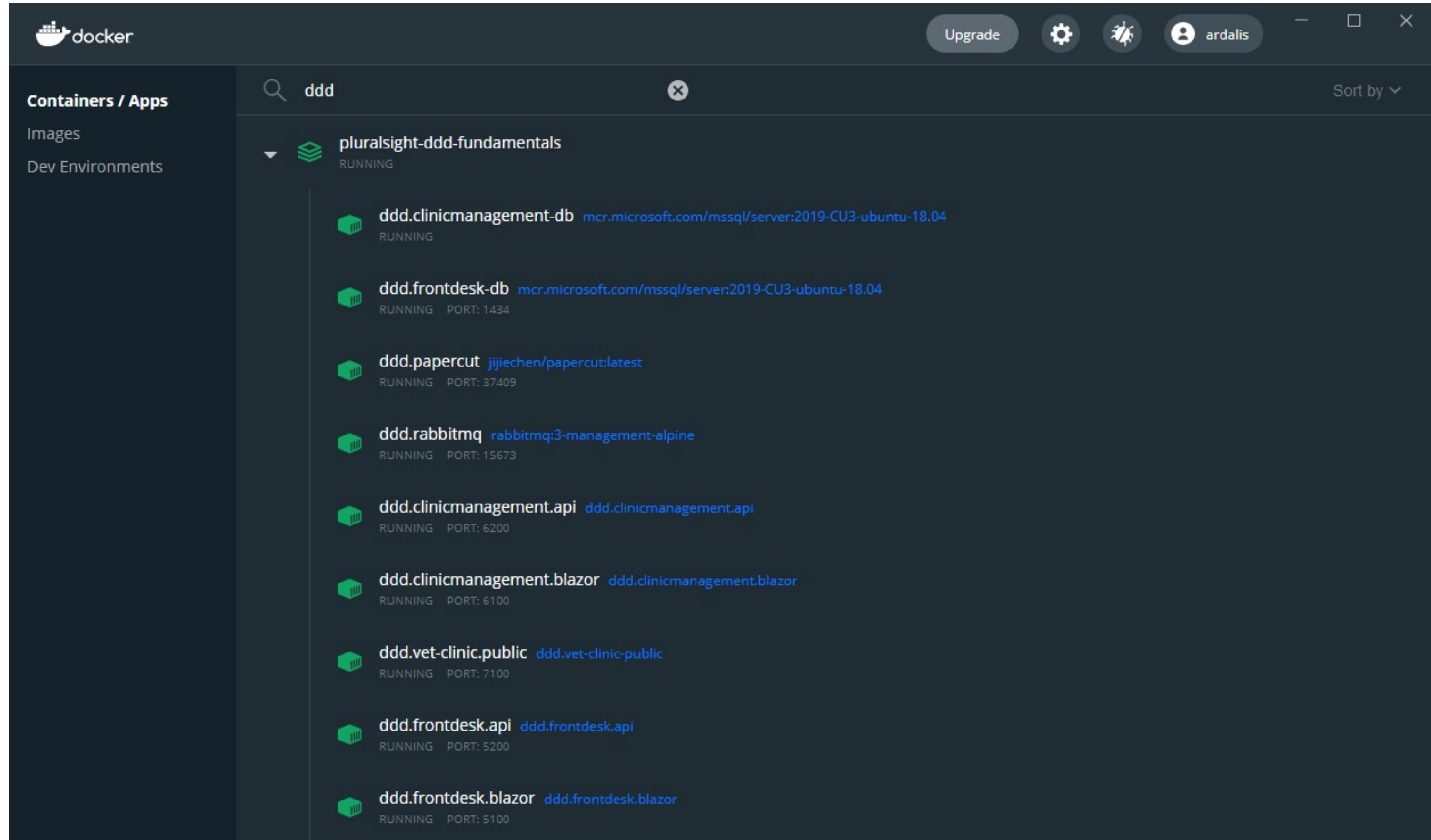
ASP.NET Core
EF Core

Docker

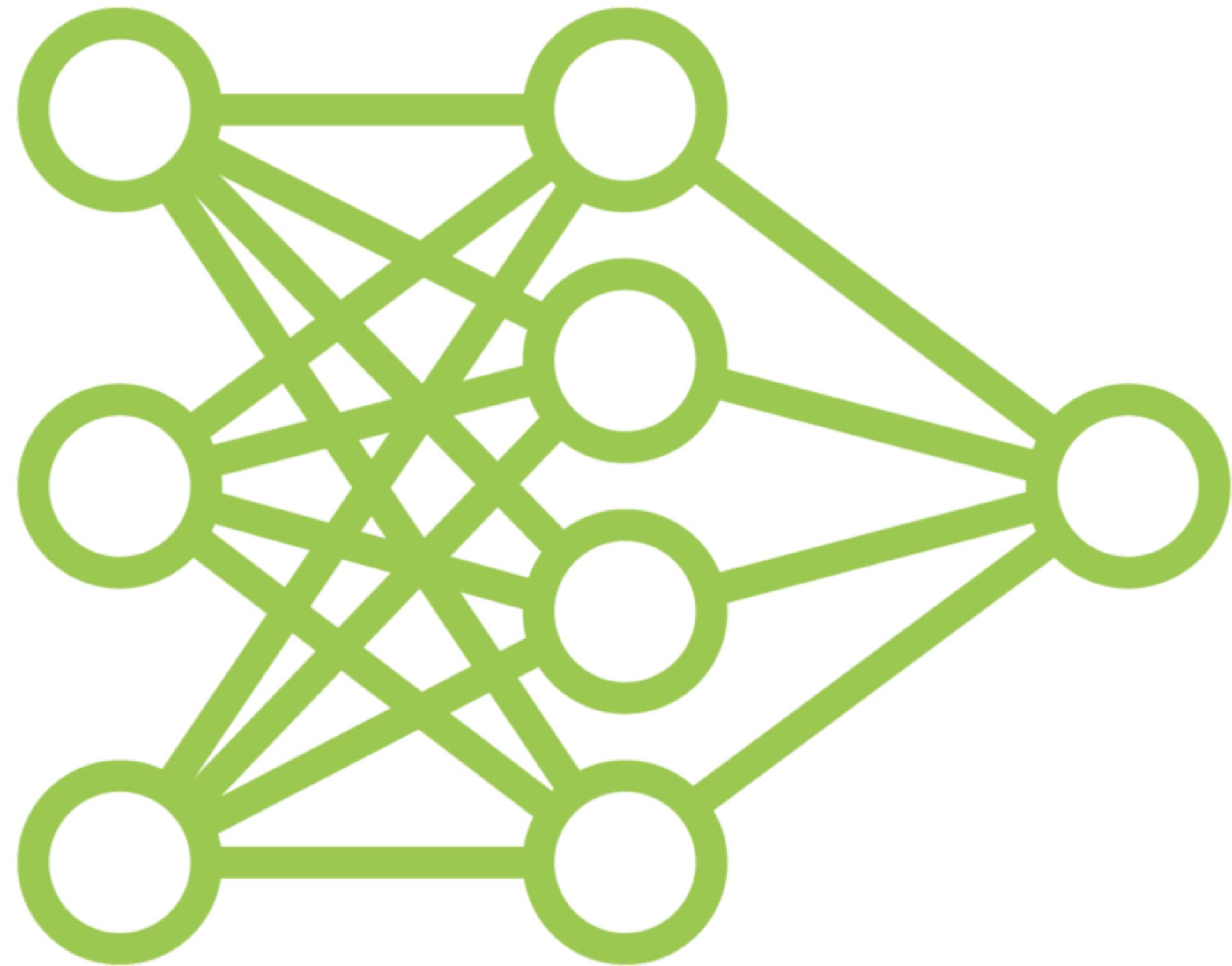
Papercut

RabbitMQ

Docker Running the Sample Apps



Microservices and DDD



EXPLORER

OPEN EDITORS

FRONDESK

- > .vs
- > .vscode
- > src
 - > BlazorShared
 - > FrontDesk.Api
 - > FrontDesk.Blazor
 - > FrontDesk.Blazor.Host
 - > FrontDesk.Core
 - > bin
 - > Constants
 - > Events
 - > Exceptions
 - > Handlers
 - > Interfaces
 - > obj
 - > ScheduleAggregate
 - > Guards
 - > Specifications
 - > C# Appointment.cs
 - > C# Schedule.cs
 - > Services
 - > SyncedAggregates
 - > ValueObjects
 - > FrontDesk.Core.csproj
 - > FrontDesk.Infrastructure
 - > tests
 - > .dockerignore
 - > .gitignore

main 54 tests 0 △ 2 .NET Core Launch (web) (FrontDesk) Live

Real World Code



Domain-Driven Design, 1st ed. (2014)

<https://bit.ly/PS-DDD>

Julie Lerman and Steve Smith

DDD is a big topic, and we're all
learning new ways to build
better software

Review and Resources

Planning and implementing
with DDD guidance made
extending the application
painless.

Key Terms from this Module

Message Queue

A tool for storing and retrieving messages, often used by applications to communicate with one another in a disconnected fashion

Service Bus

Software responsible for managing how messages are routed between numerous applications and services

Key Takeaways



Previous design set us up for easily adding features

Used a message queue (RabbitMQ) to communicate between apps

Leveraged 3rd party products for generic domains

Debugged through code to see how the data and events flowed

Highlighted modeling tools & practices and new ideas emerging from DDD community

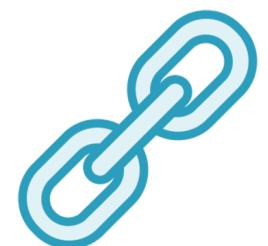
Shared lessons learned since first iteration of this course!



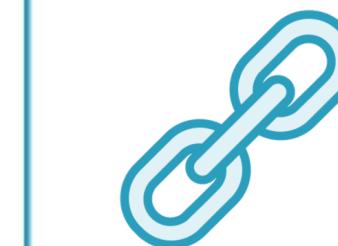
Thanks to Eric Evans for
his great advice and insights

domainlanguage.com

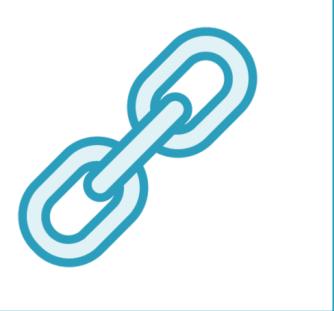
Resources Referenced in This Module



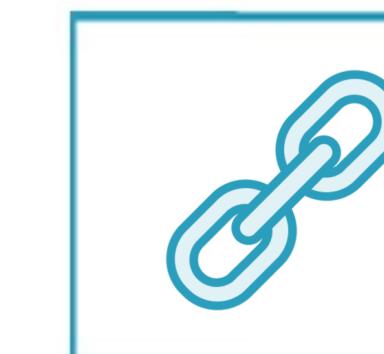
Bus or Queue (Blog post):
ardalis.com/bus-or-queue



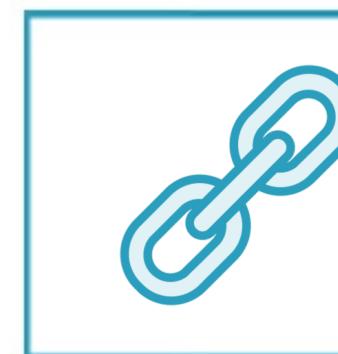
EventModeling.com



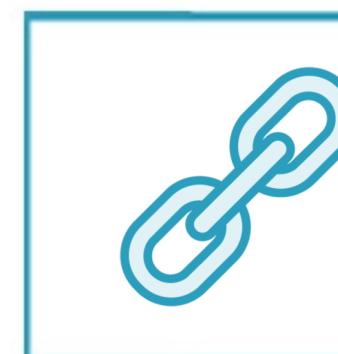
RabbitMQ.com



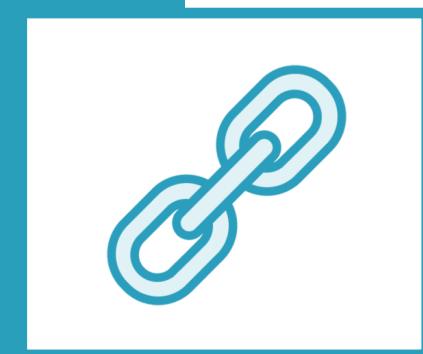
DomainStorytelling.org



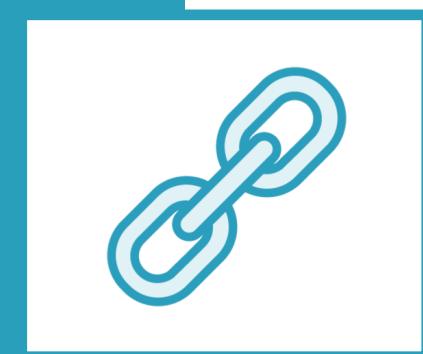
Domain Storytelling Modeler
wps.de/modeler



Bounded Context Canvas

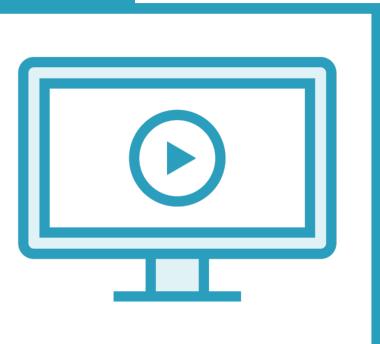


Miro.com

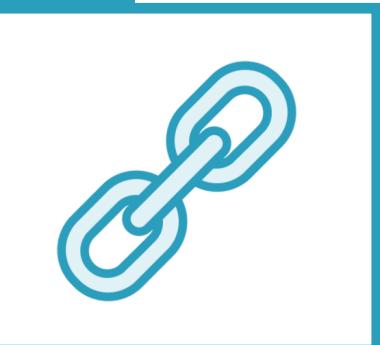


EventStorming.org

More Resources Referenced in This Module



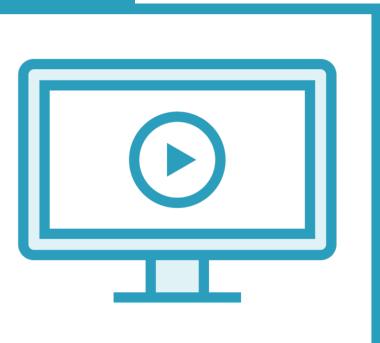
Developing Core Business Applications with Domain-Driven Design (DDD) and Microsoft .NET
Jimmy Nilsson, TechEd bit.ly/NilssonDDTechEd2013



Papercut github.com/ChangemakerStudios/Papercut-SMTP



RabbitMQ By Example, Stephen Haunts on Pluralsight
app.pluralsight.com/library/courses/rabbitmq-by-example



A Clean Architecture Solution Template for .NET Developers
github.com/ardalis/CleanArchitecture

Domain-Driven Design Fundamentals



Steve Smith
Force Multiplier for
Dev Teams
[@ardalis](https://twitter.com/ardalis) ardalis.com



Julie Lerman
Software coach,
DDD Champion
[@julielerman](https://twitter.com/julielerman) thedatafarm.com