Learning Event Storming for Rapid Business Process Discovery

Jacob See
Senior Consulting Engineer

Haitham Shahin
Senior Consulting Engineer

Key Takeaways

Understand Domain-Driven Design

What/Why/How on Event Storming

Event Storming $\longrightarrow$ Development

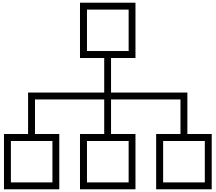
What's your experience with Event Storming / DDD / Other Practices?

Domain-Driven Design is a *language* and *domain-centric* approach to software design for complex domains

Complexity from the domain is **inherent**

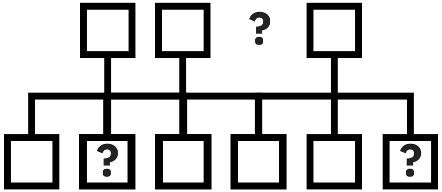
Complexity from the technical solution is **accidental**

1



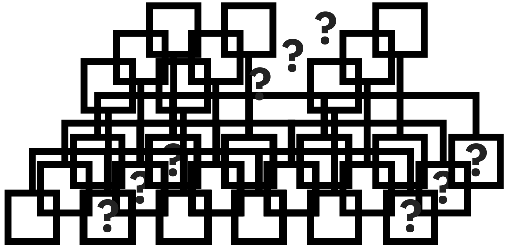
Initial product is fast to create

2



Complexity increases over time without conscious effort to organize & mitigate it

3

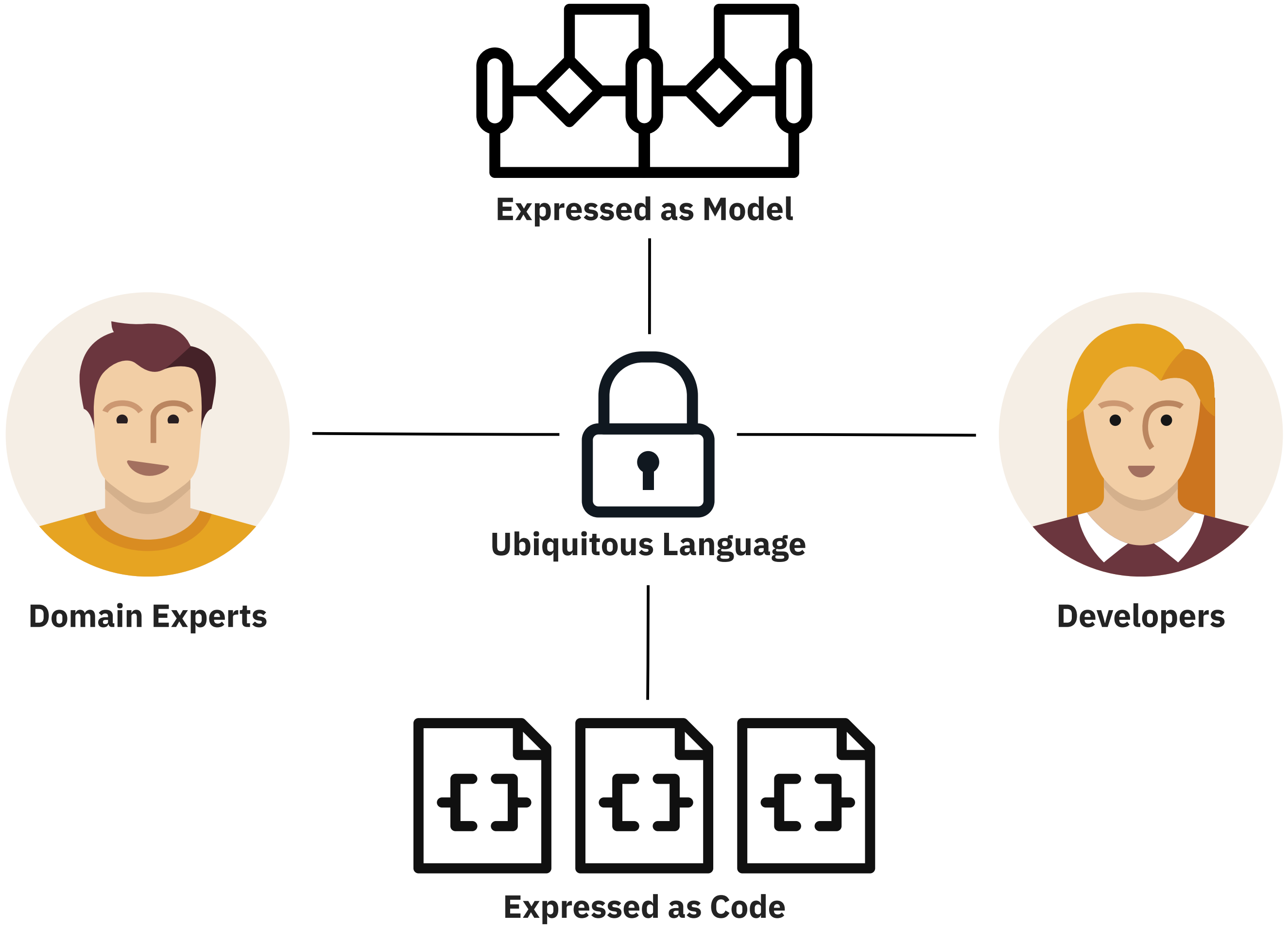


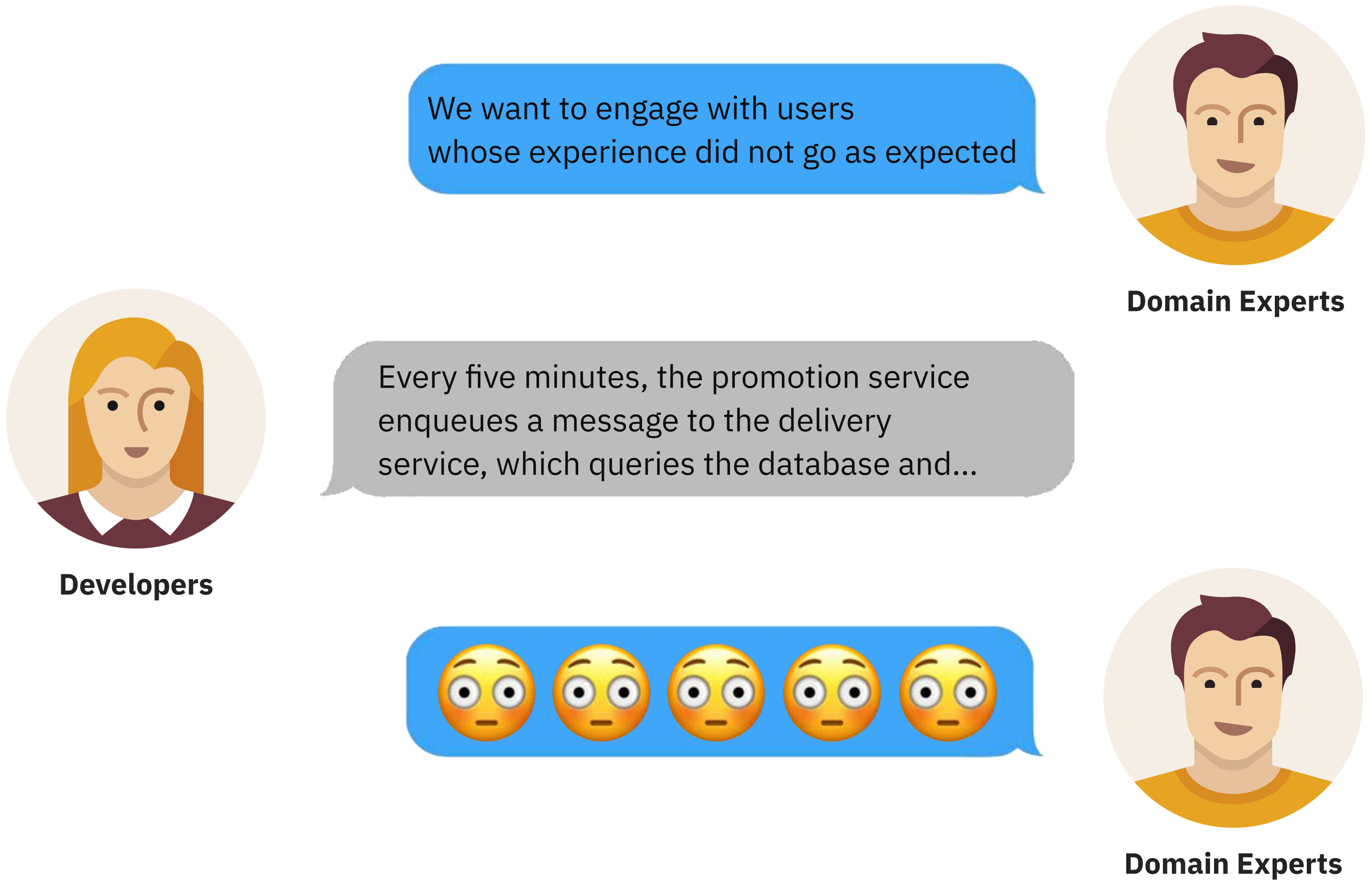
It *works* (technically...)

Change is risky

WHAT'S GOING ON ANYMORE?

Understanding the system is overly time-consuming

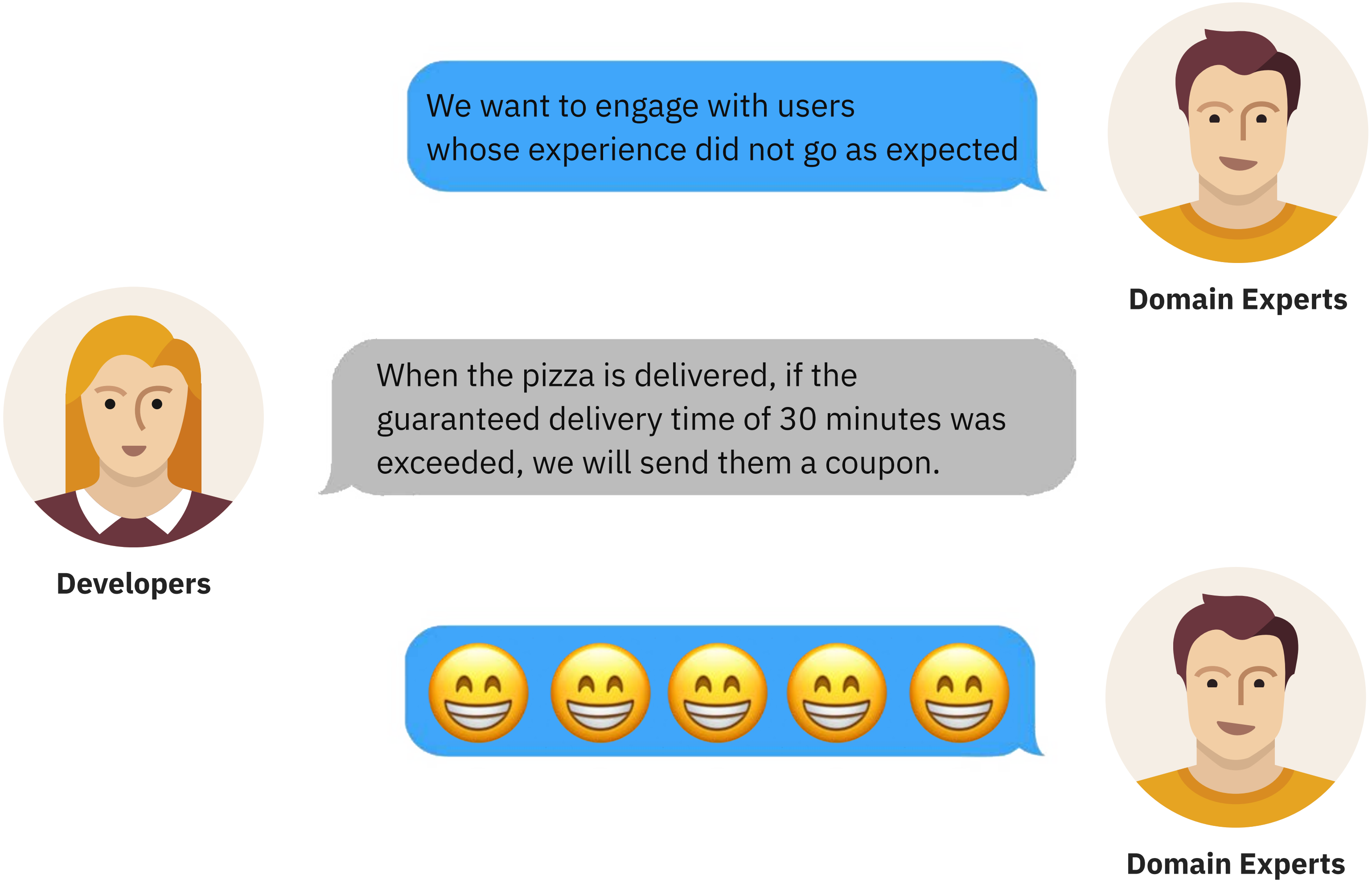

Expressed as Model
Domain Experts
Ubiquitous Language
Developers
Expressed as Code



We want to engage with users whose experience did not go as expected

Every five minutes, the promotion service enqueues a message to the delivery service, which queries the database and...

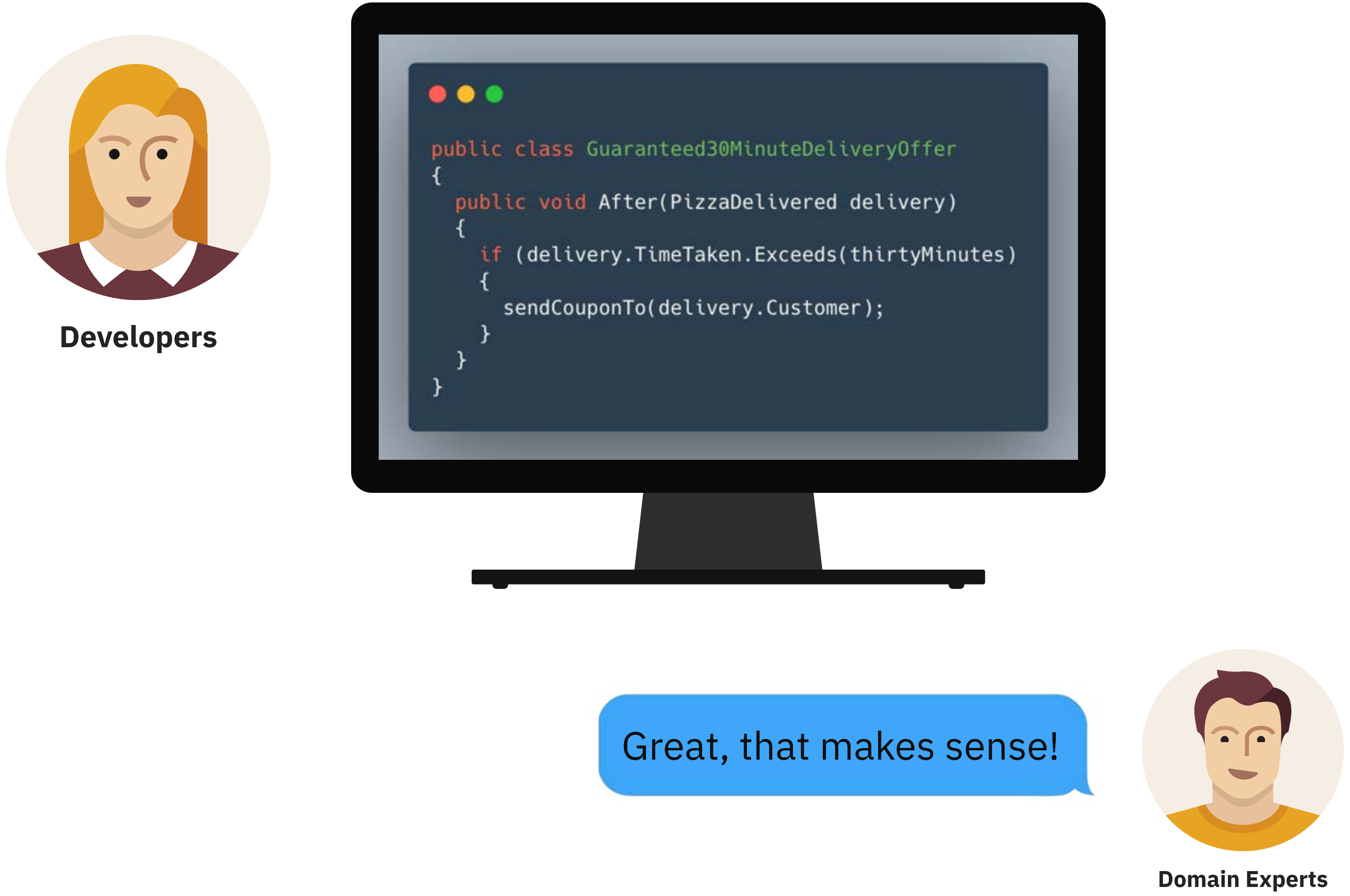
😳😳😳😳😳



We want to engage with users whose experience did not go as expected

When the pizza is delivered, if the guaranteed delivery time of 30 minutes was exceeded, we will send them a coupon.

😁😁😁😁😁



```
public class Guaranteed30MinuteDeliveryOffer
{
   public void After(PizzaDelivered delivery)
   {
      if (delivery.TimeTaken.Exceeds(thirtyMinutes)
      {
         sendCouponTo(delivery.Customer);
      }
   }
}
```

Event Storming is a **collaborative** approach to modeling business processes that follows **DDD** methodology

Expressed as Model

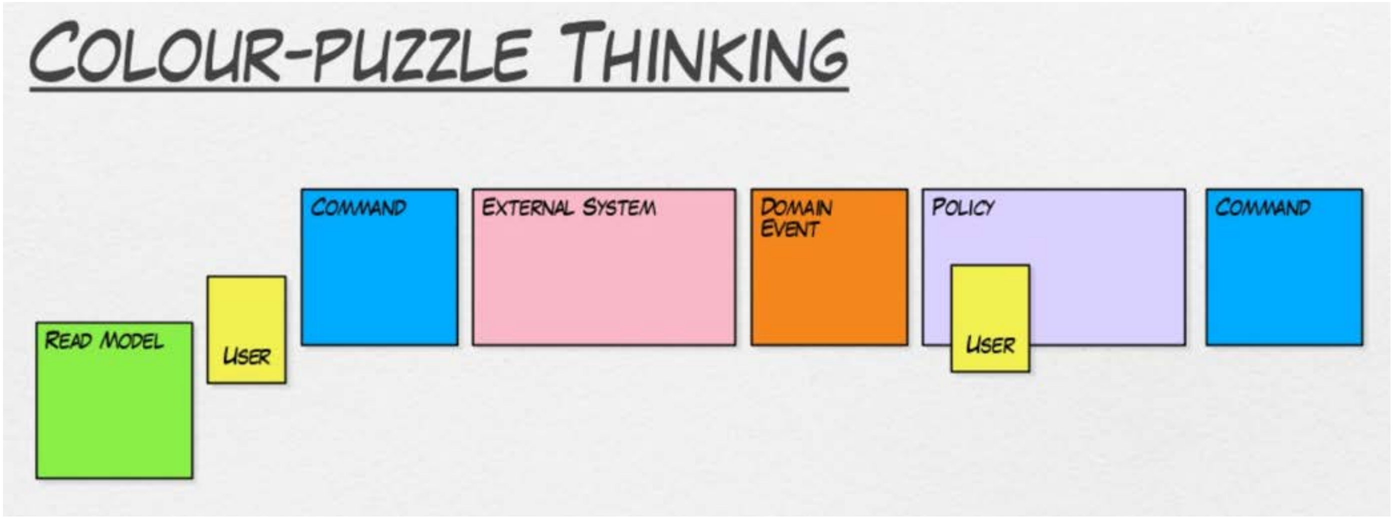
Domain Experts

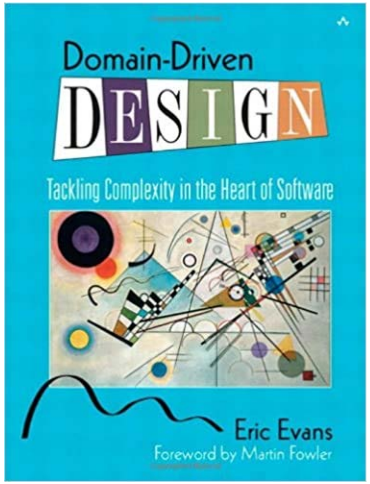
Ubiquitous Language

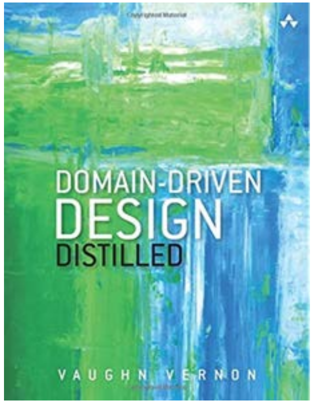
Developers

Expressed as Code

COMMANDS

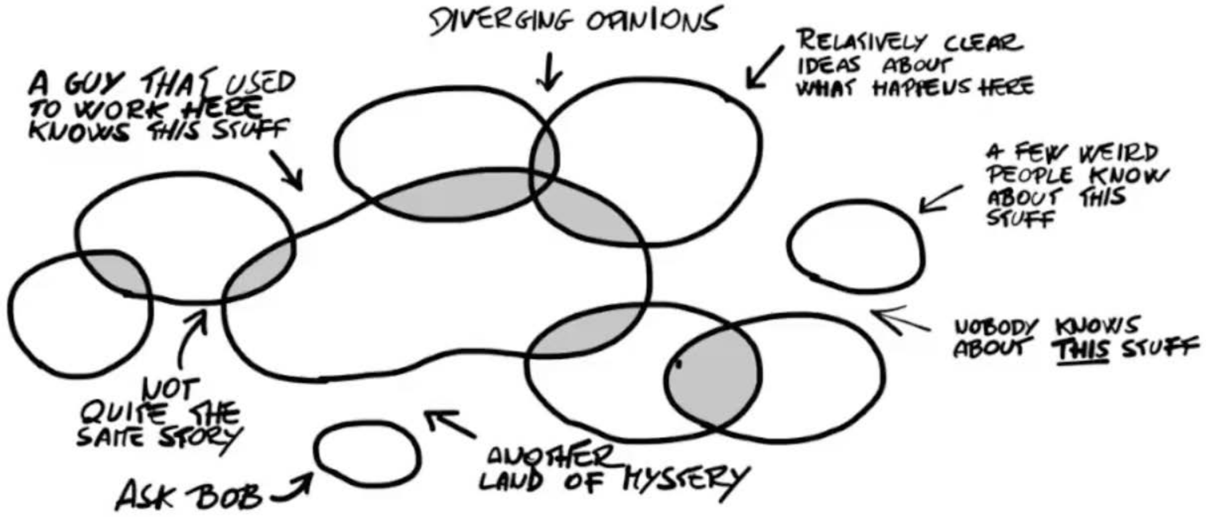

COLOUR-PUZZLE THINKING
READ MODEL
USER
COMMAND
EXTERNAL SYSTEM
DOMAIN EVENT
POLICY
USER
COMMAND


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

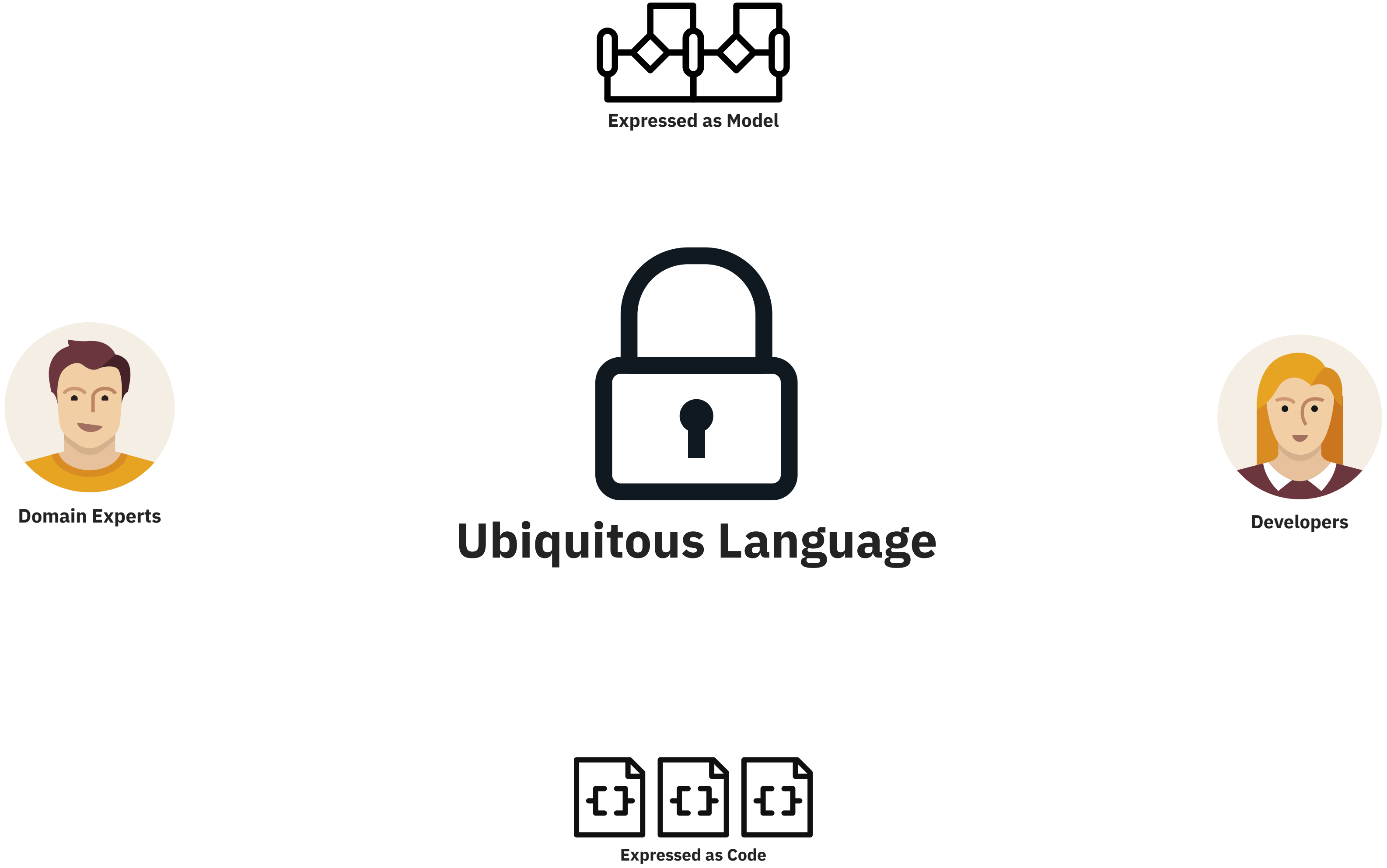


Alberto Brandolini

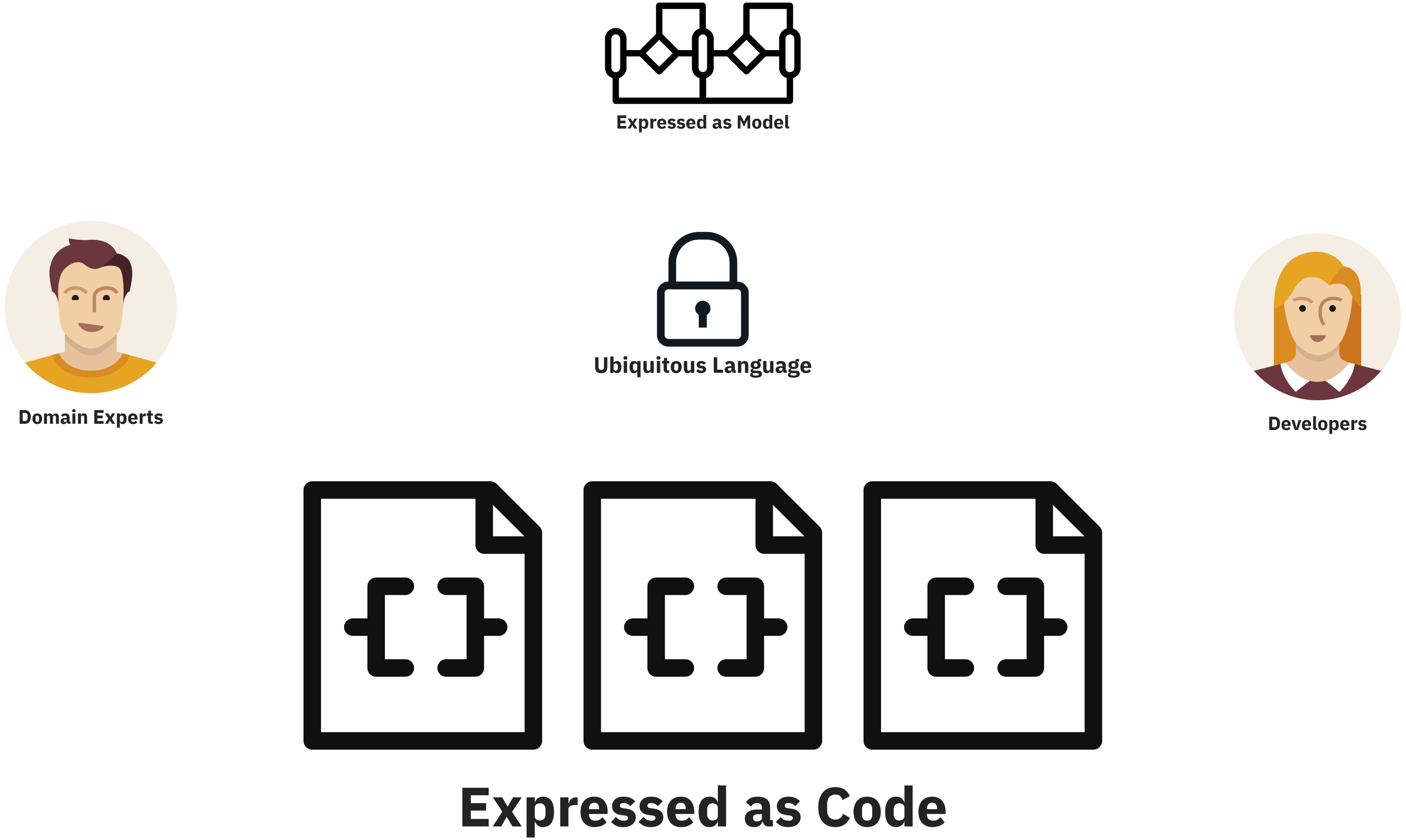
THE KNOWLEDGE DISTRIBUTION


DIVERGING OPINIONS
RELATIVELY CLEAR IDEAS ABOUT WHAT HAPPENS HERE
A GUY THAT USED TO WORK HERE KNOWS THIS STUFF
A FEW WEIRD PEOPLE KNOW ABOUT THIS STUFF
NOBODY KNOWS ABOUT THIS STUFF
NOT QUITE THE SAME STORY
ASK BOB
ANOTHER LAND OF MYSTERY

Create **Alignment** and Shared Understanding of Problem Space


Expressed as Model
Domain Experts
Ubiquitous Language
Developers
Expressed as Code

Model **Event-Driven** Systems and Discover **Microservices**


Expressed as Model
Domain Experts
Ubiquitous Language
Developers
Expressed as Code

Demo Time!

Customer Service Inbound Call Handling

Can you help me with my account balance?

Customer

Handler

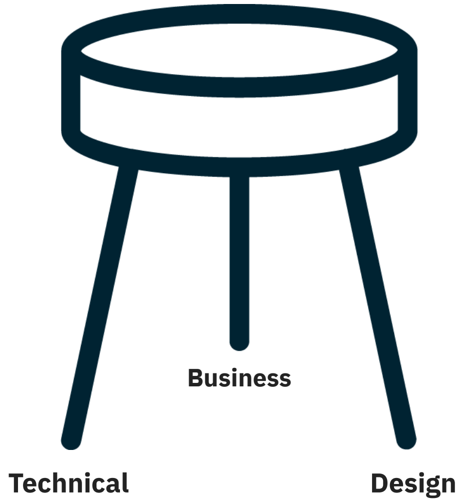
Your account balance is $1,000.00. Anything else I can help with today?

No, thank you for your help!

Customer

Who is in the room?

Event Storm Facilitators (us!)



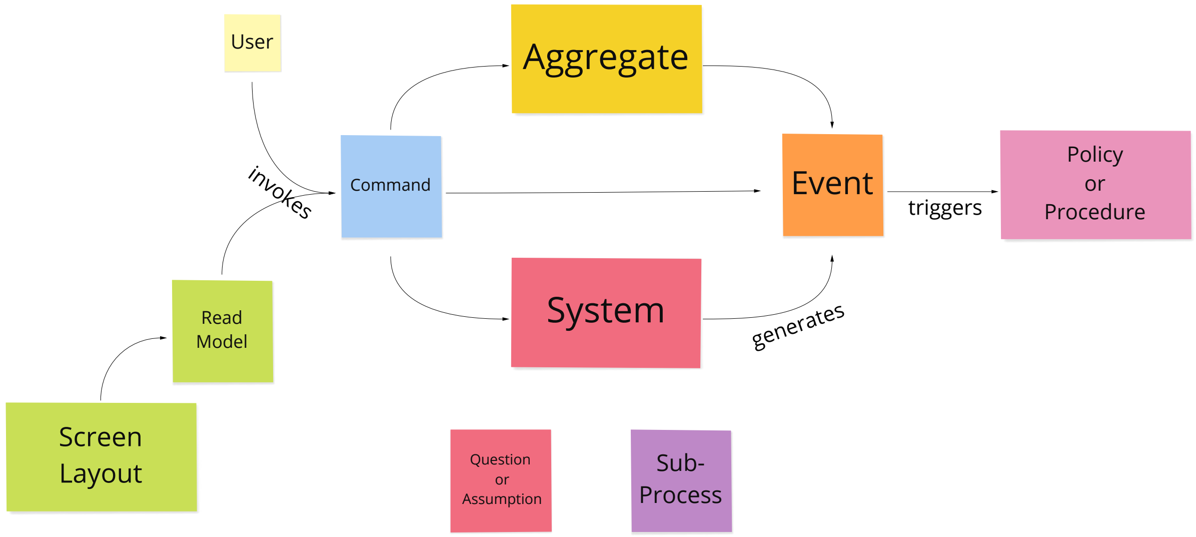
Business Stakeholders (Domain Experts)

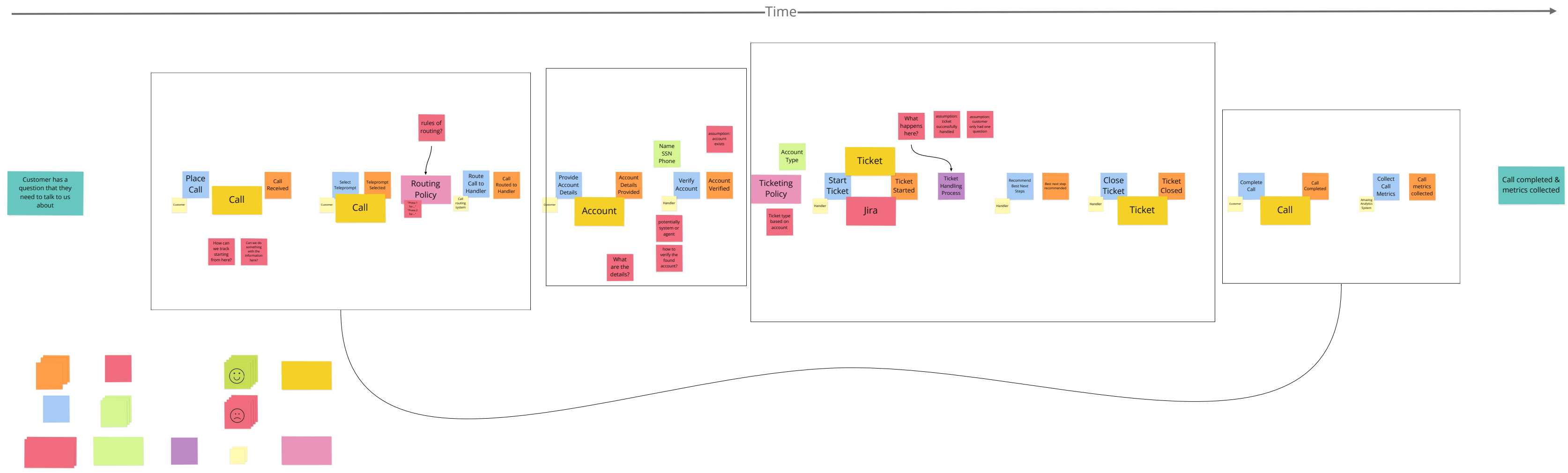
Product Owner

Developers/Architects

UI/UX

Key


User
Aggregate
Command
invokes
Event
triggers
Policy or Procedure
Read Model
System
generates
Screen Layout
Question or Assumption
Sub-Process


Time
Customer has a question that they need to talk to us about
Place Call
Customer
Call
Call Received
How can we track starting from here?
Can we do something with the information here?
Select Teleprompt
Customer
Call
Teleprompt Selected
rules of routing?
Routing Policy
"Press 1 for..." "Press 2 for..."
Call routing system
Route Call to Handler
Call Routed to Handler
Provide Account Details
Customer
Account
Account Details Provided
What are the details?
Name SSN Phone
Verify Account
Handler
potentially system or agent
how to verify the found account?
assumption: account exists
Account Verified
Account Type
Ticketing Policy
Ticket type based on account
Start Ticket
Handler
Ticket
Jira
Ticket Started
What happens here?
assumption: ticket successfully handled
assumption: customer only had one question
Ticket Handling Process
Recommend Best Next Steps
Handler
Best next step recommended
Close Ticket
Handler
Ticket
Ticket Closed
Complete Call
Customer
Call
Call Completed
Collect Call Metrics
Amazing Analytics System
Call metrics collected
Call completed & metrics collected

Events

something that happened that people care about

<subject> <verb> (always past tense!)

order submitted

item shipped

could be timed

-or-

the meaningful result of another thing

nightly accounts reconciled

account locked (wrong password 3x)

where do events come from?

→ could be a system

→ the passing of TIME

→ or the consequence of another event

Commands

! an action started by an actor

it represents the decision

cancel reservation

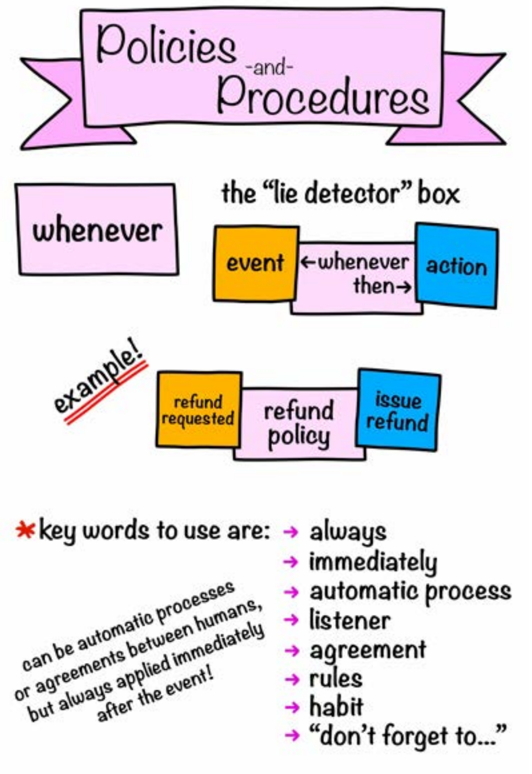
usually the reverse of the event

Actors

the "user" of the system

* keep it fuzzy *

the actor makes the decision



Policies -and- Procedures

whenever

the "lie detector" box

example!

*key words to use are:
- → always
- → immediately
- → automatic process
- → listener
- → agreement
- → rules
- → habit
- → "don't forget to..."

can be automatic processes or agreements between humans, but always applied immediately after the event!

Sub-Process

Represents a distinct process/model

Keeps this event storm focused

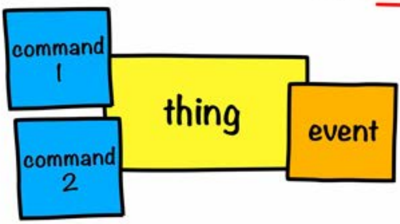
Separate event storm to model subprocess

Aggregates

aggregate part of the system that receives the command and decides to execute the event

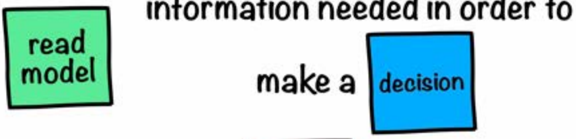
notification
order
usually a noun

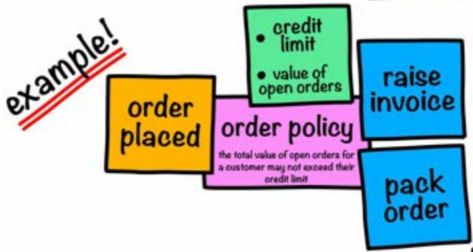
* the aggregate is the <u>state machine</u>

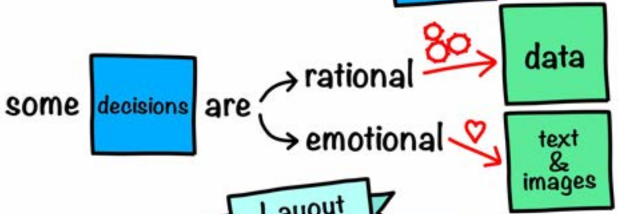


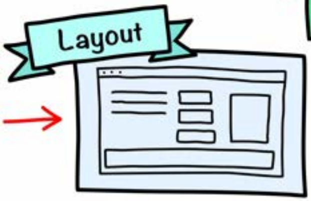
they logically group commands once all event sources are defined

Read Model









ties to the layout

Questions

- pain points
- how / what / why
- hot spots
- assumptions

this bit takes ages!
where does this data come from?
Jeff does this
no one really understands this
?
could be this thing or that thing

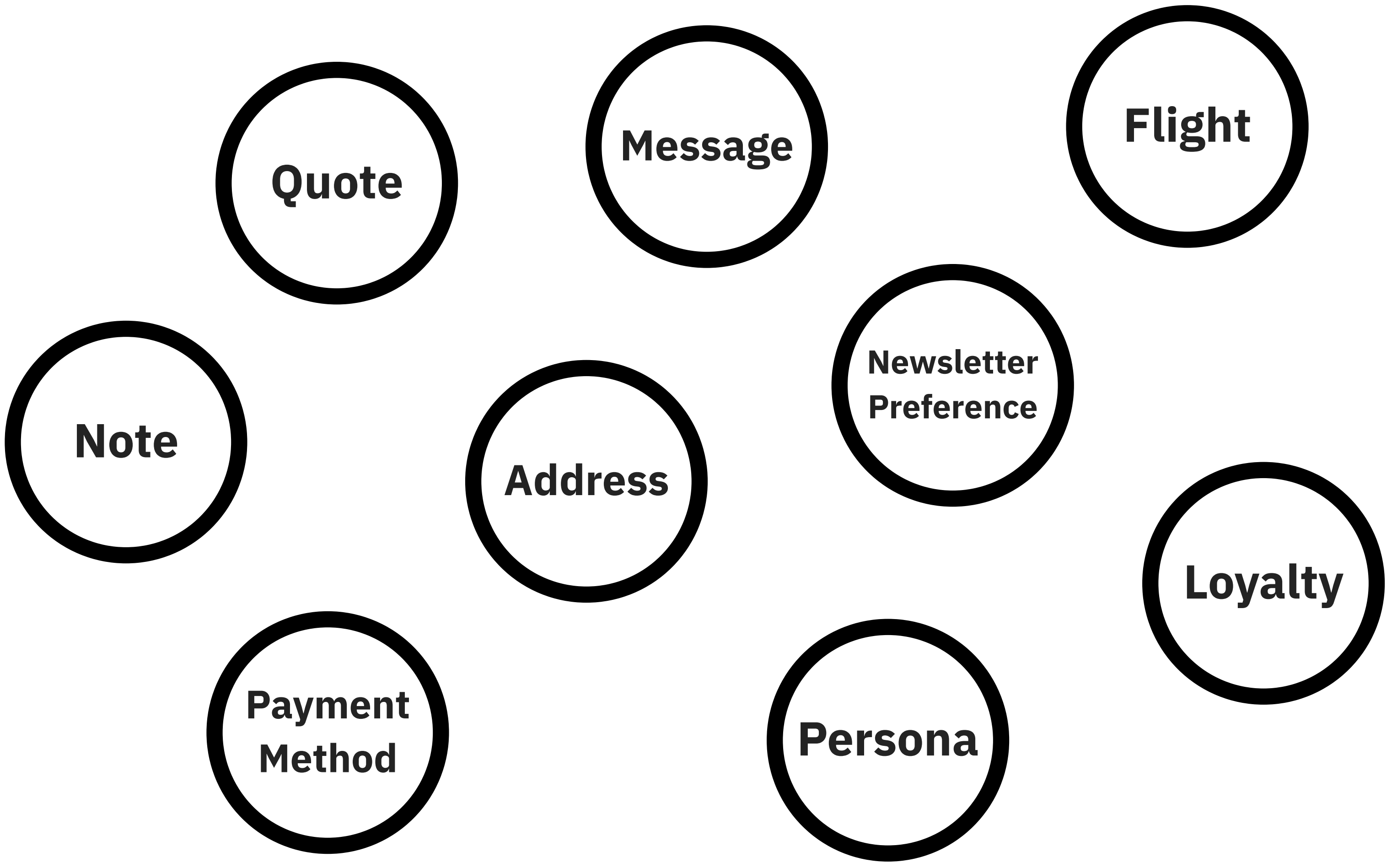
visualize both sides of the argument

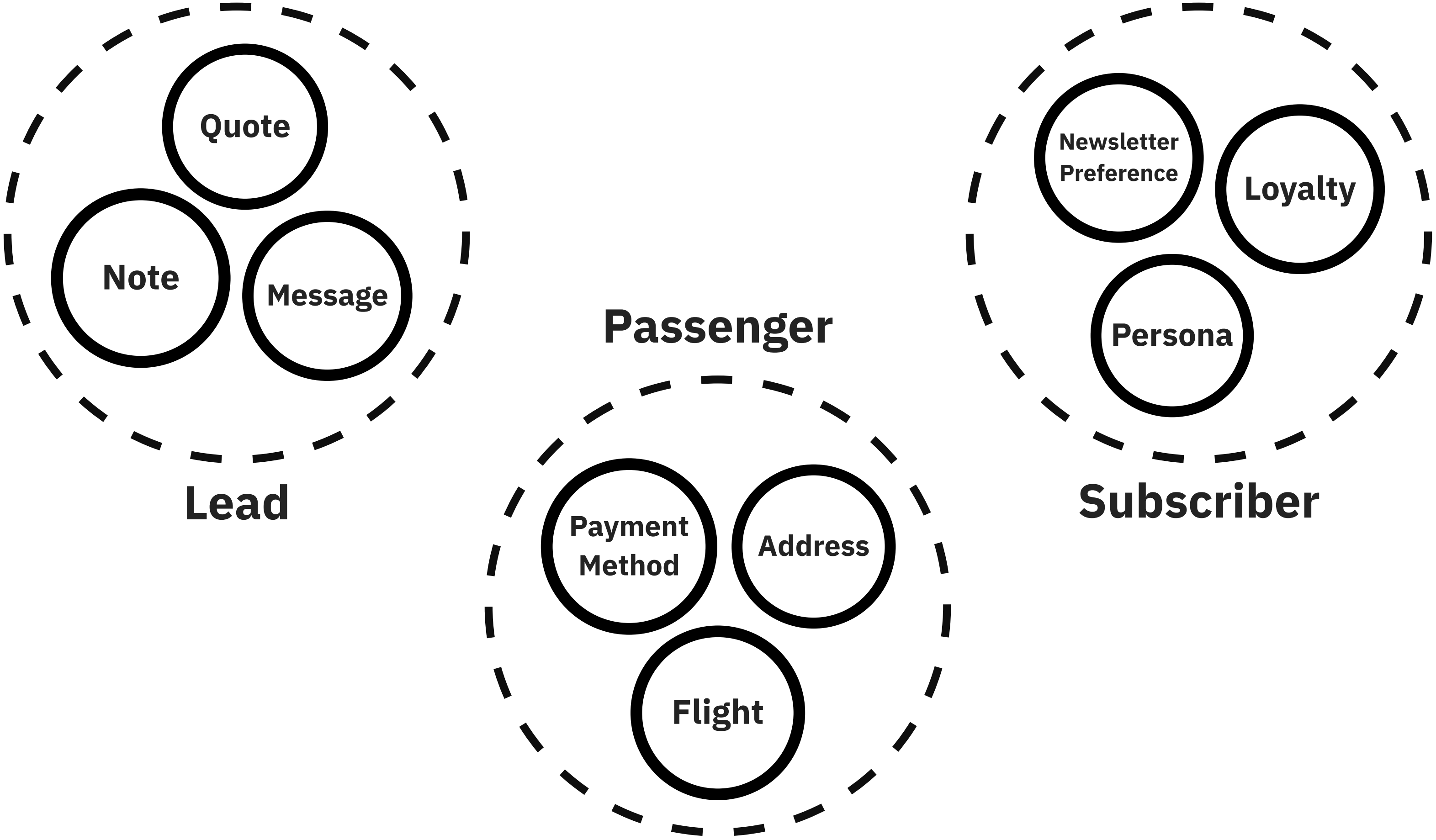
External Systems

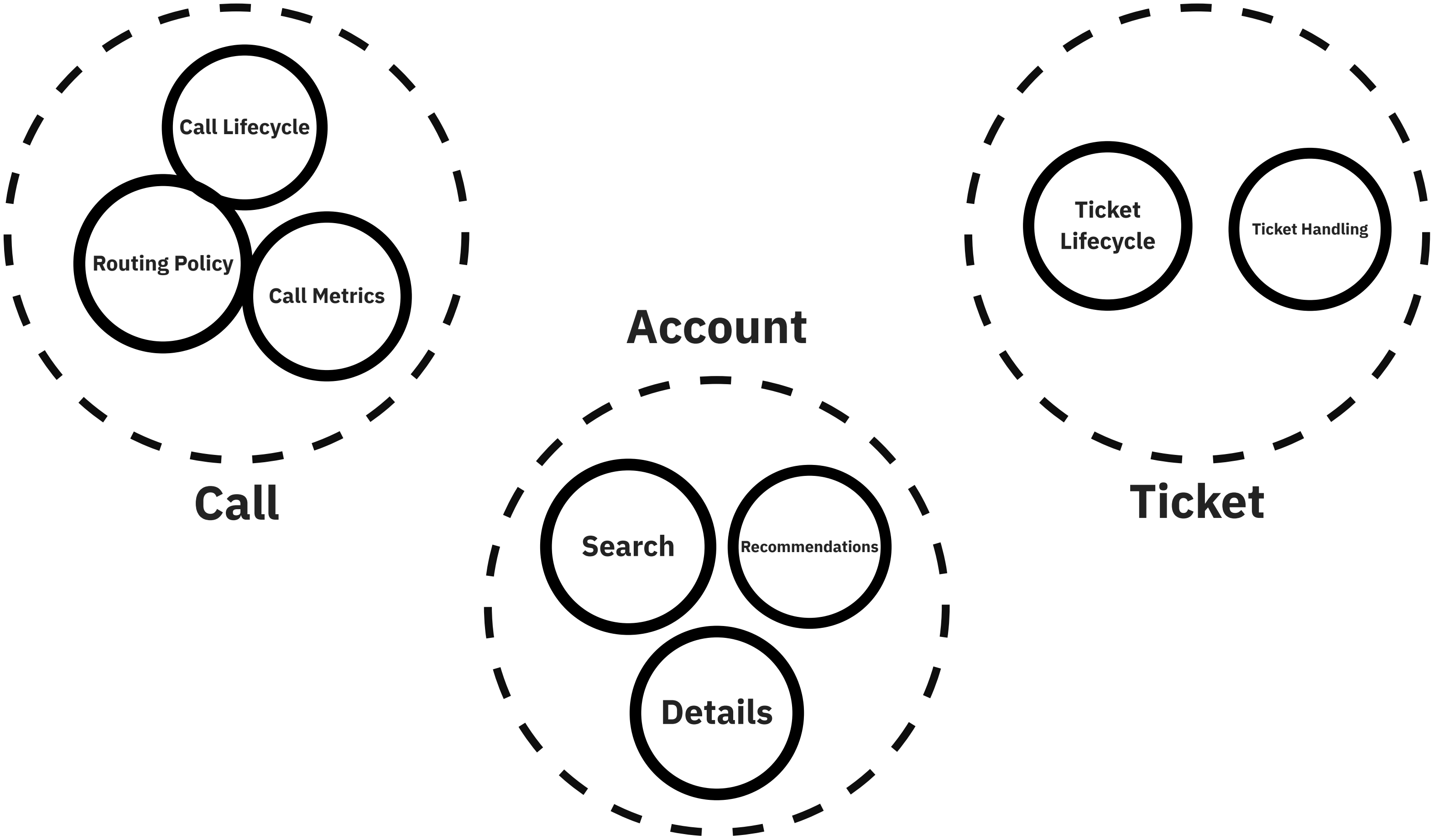
- third-party services
- existing systems

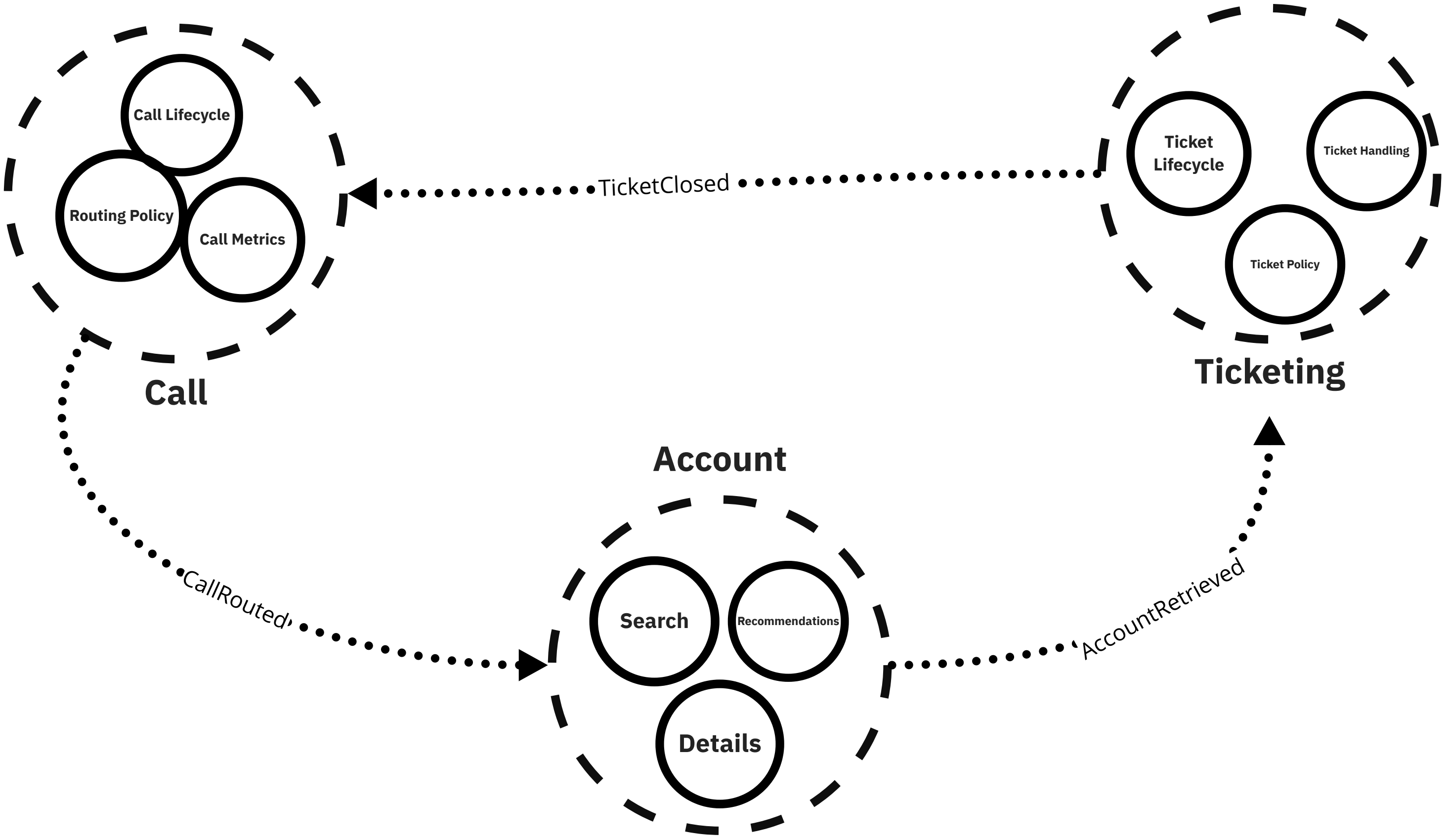
system
payment provider
mail server

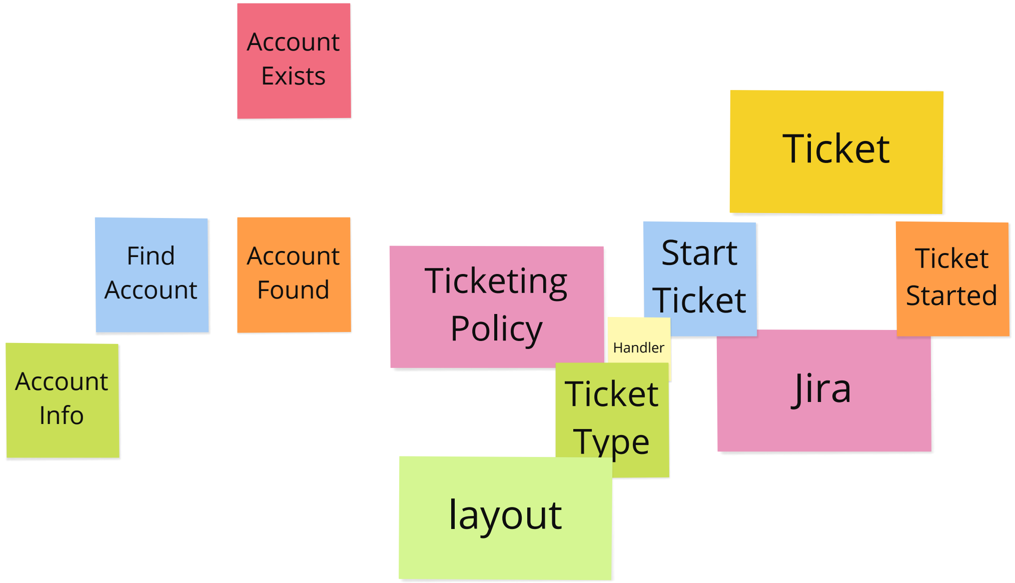
Bounded Contexts


Quote
Message
Flight
Note
Address
Newsletter Preference
Loyalty
Payment Method
Persona


Quote
Note
Message
Lead
Passenger
Payment Method
Address
Flight
Newsletter Preference
Loyalty
Persona
Subscriber


Call Lifecycle
Routing Policy
Call Metrics
Call
Account
Search
Recommendations
Details
Ticket Lifecycle
Ticket Handling
Ticket


Call Lifecycle
Routing Policy
Call Metrics
Call
TicketClosed
Ticket Lifecycle
Ticket Handling
Ticket Policy
Ticketing
CallRouted
Account
Search
Recommendations
Details
AccountRetrieved


Account Exists
Ticket
Find Account
Account Found
Ticketing Policy
Start Ticket
Ticket Started
Handler
Account Info
Ticket Type
Jira
layout

User Story

As an <ACTOR>

I want <COMMAND>

So that <REASON/VALUE>

As a handler

I want to create a ticket in Jira

So that we can audit requests

User Story

As an <ACTOR>

I want <COMMAND>

So that <REASON/VALUE>

As a handler

I want to create a ticket in Jira

So that we can audit requests

Acceptance Criteria

Given <ASSUMPTIONS/PAST EVENTS>

When <COMMAND>

Then <EVENT>

Given an account has been retrieved

When a handler creates a ticket

Then a TicketCreated event is issued

User Story

As an <ACTOR>

I want <COMMAND>

So that <REASON/VALUE>

As a handler

I want to create a ticket in Jira

So that we can audit requests

Acceptance Criteria

Given <ASSUMPTIONS/PAST EVENTS>

When <COMMAND>

Then <EVENT>

Given an account has been retrieved

When a handler creates a ticket

Then a TicketCreated event is issued

Given <ASSUMPTIONS/PAST EVENTS/POLICY>

When <COMMAND>

Then <EVENT>

Given an account has been retrieved

And the account is a premium account

When a handler creates a ticket

Then a premium ticket is created

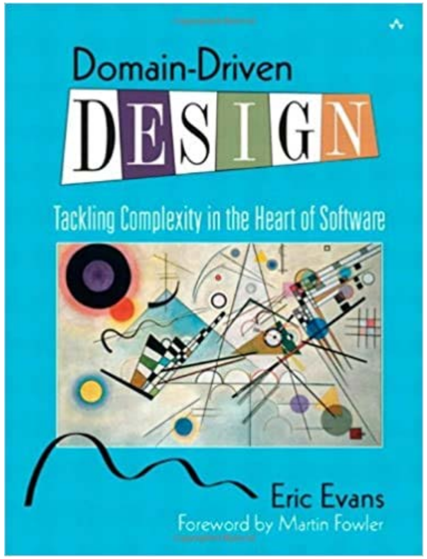
And a TicketCreated event is issued

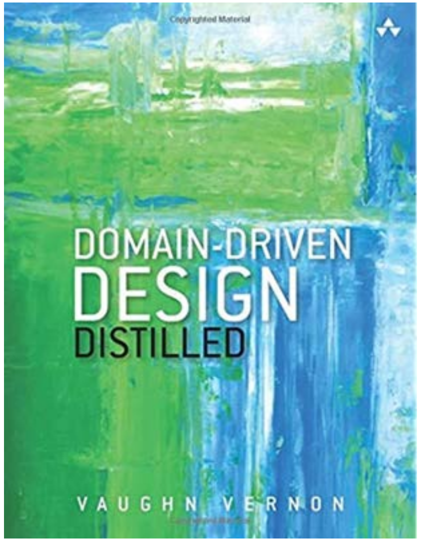
Key Takeaways

Understand Domain-Driven Design

What/Why/How on Event Storming

Event Storming $\longrightarrow$ Development


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler



Eric Evans

Vaughn Vernon

Alberto Brandolini

www.eventstorming.com

EventStorming

The smartest approach to collaborate beyond silo boundaries.

Credit for some slides/diagrams/ideas!

openpracticelibrary.com

Event Storming

What is Event Storming? Event Storming is a rapid, interactive approach to business process discovery and design that yields high quality...

Questions

Jacob See
Senior Consulting Engineer

Haitham Shahin
Senior Consulting Engineer

Keep Iterating!