

Hustle and Flow

“How do we design flows to deliver asynchronous conversations”

Ian Cooper

@ICooper

Who are you?

- Software Developer for more than 25 years
 - Stuff I care about: Messaging, EDA, Microservices, TDD, XP, OO, RDD & DDD, Code that Fits in My Head, C#
 - Places I have worked: DTI, Reuters, Sungard, Beazley, Huddle, Just Eat Takeaway
- No smart folks
 - Just the folks in this room



Welcome to Brighter

This project is a Command Processor & Dispatcher implementation with support for task queues that can be used as a lightweight library.

It can be used for implementing [Ports and Adapters](#) and [CQRS \(PDF\)](#) architectural styles in .NET.

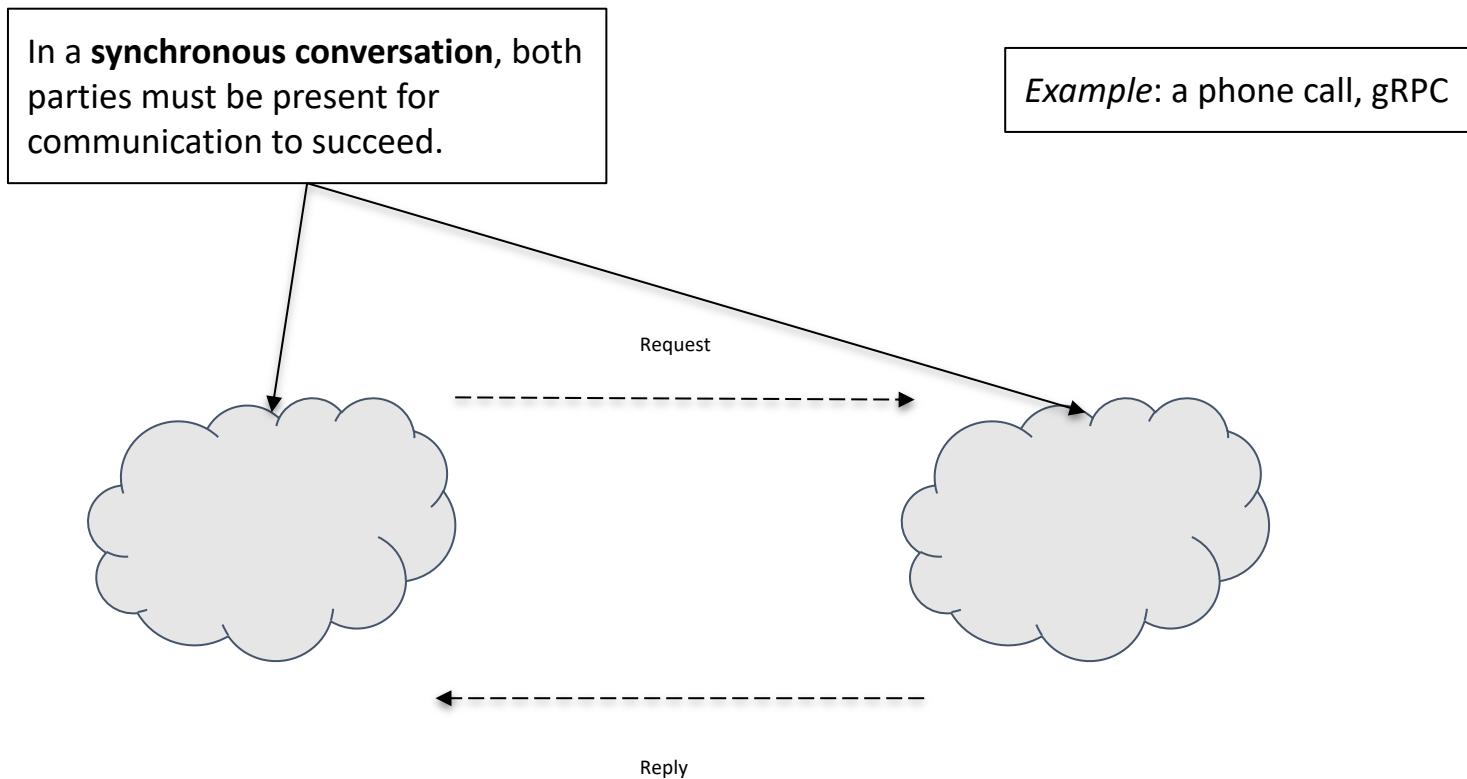
It can also be used in microservices architectures for decoupled communication between the services

[GET STARTED](#)

Agenda

- Synchronous and Asynchronous Conversations
- Paper Workflows
- Flow-Based Programming
- Hustle and Flow

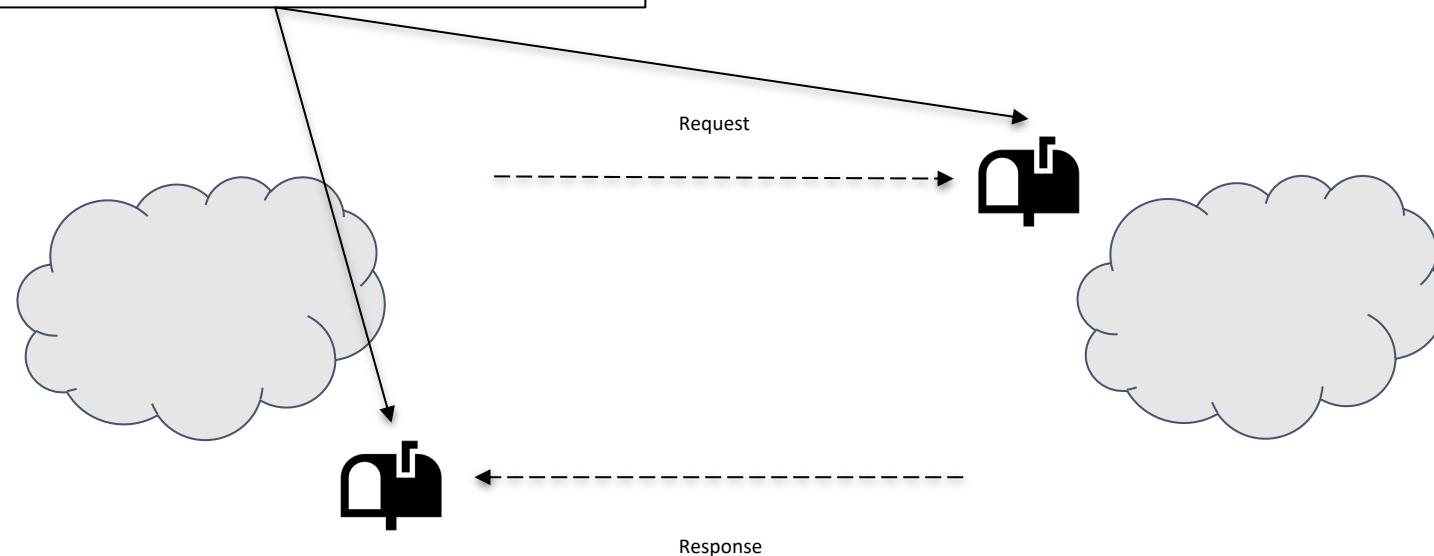
Hustle - Synchronous Conversation



Flow - Asynchronous Conversation

In an **asynchronous conversation**, the *receiver* does not need to be present at the time the *sender* communicates with them, using **store and forward** to pick up the message later.

Example: snail mail, Kafka

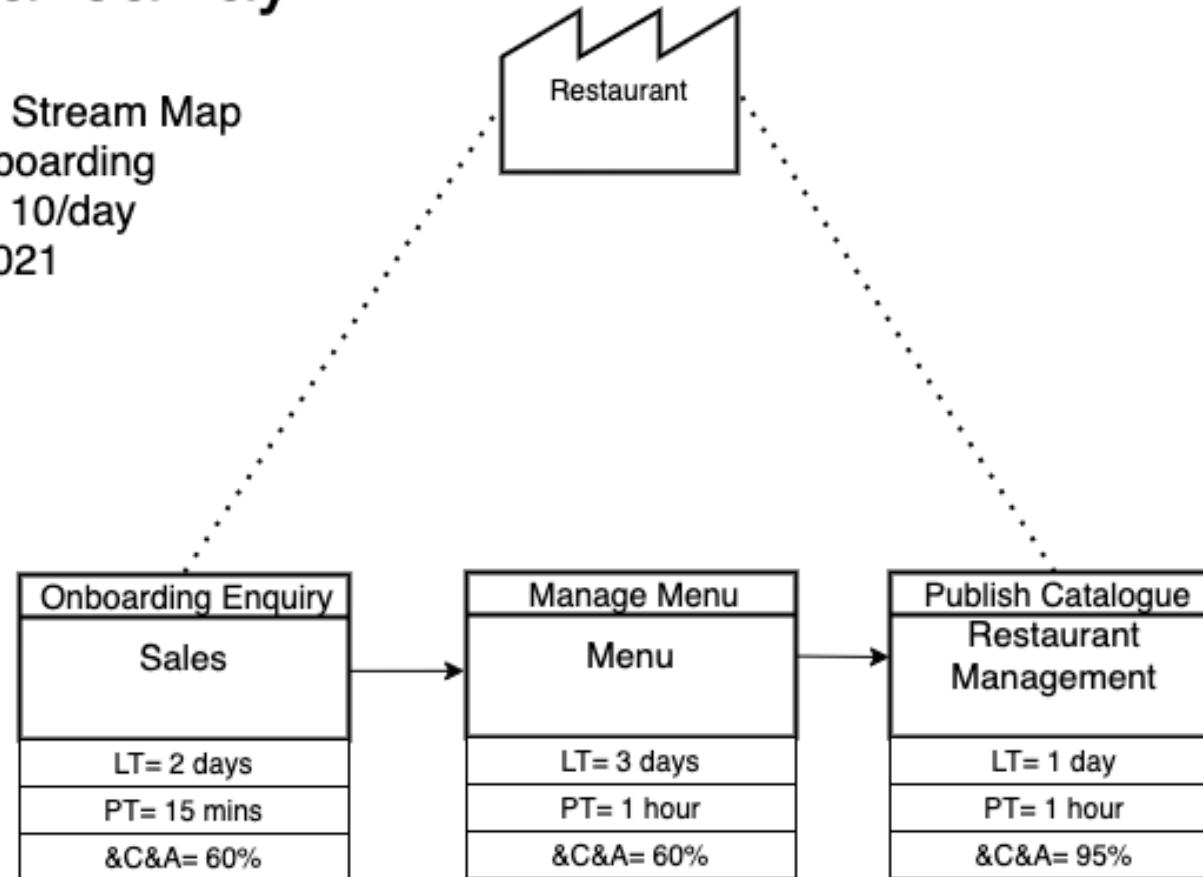


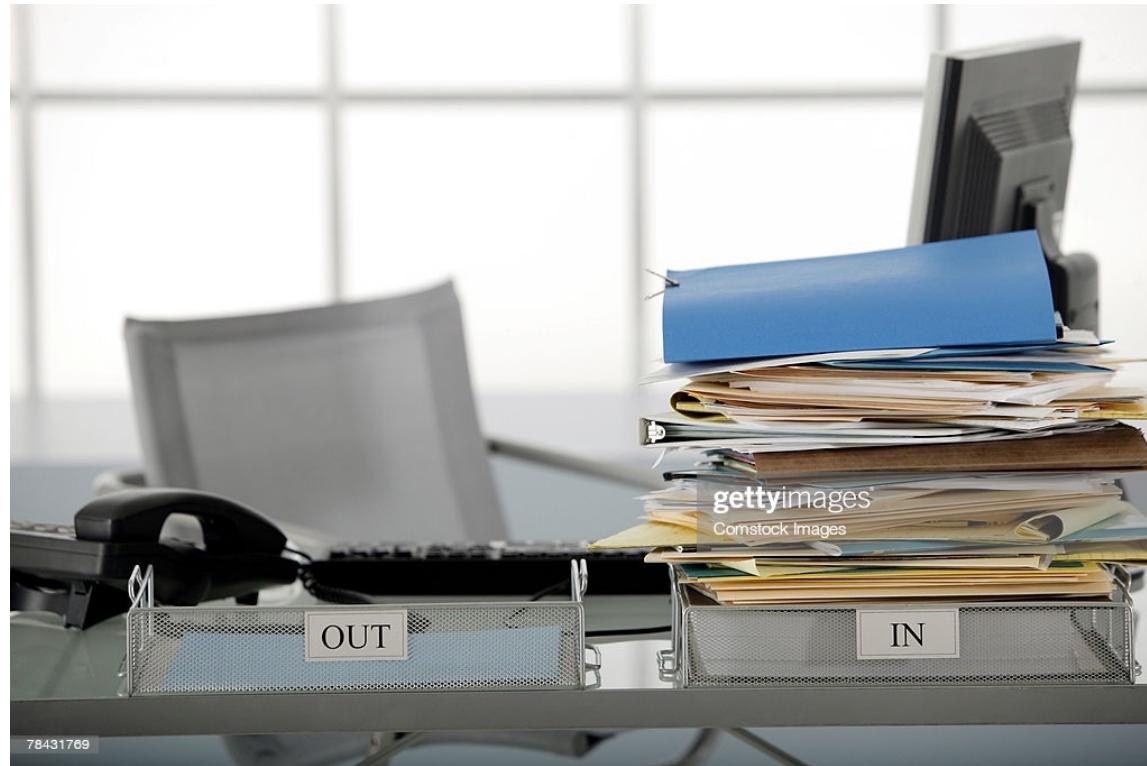
Paper Workflows

“My life looked good on paper - where, in fact, almost all of it was being lived.” - Martin Amis

Just Paper Takeaway

Current State Value Stream Map
Restaurant Onboarding
Demand Rate 10/day
29 SEP 2021







(Message)Event – Skinny, Notification
Discrete, Immediately Actioned



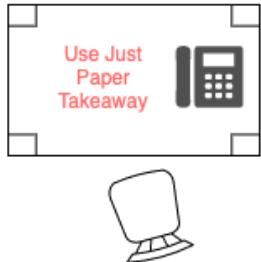
(Message)Document – Fat
Series, Supports Action





1

Restaurant
Owner



O E DIVISION
O E D I ROUTING SHEET

TITLE: EID-E15 (23 July 84)

REQD legend: I = Information
A = Action
R = Retain

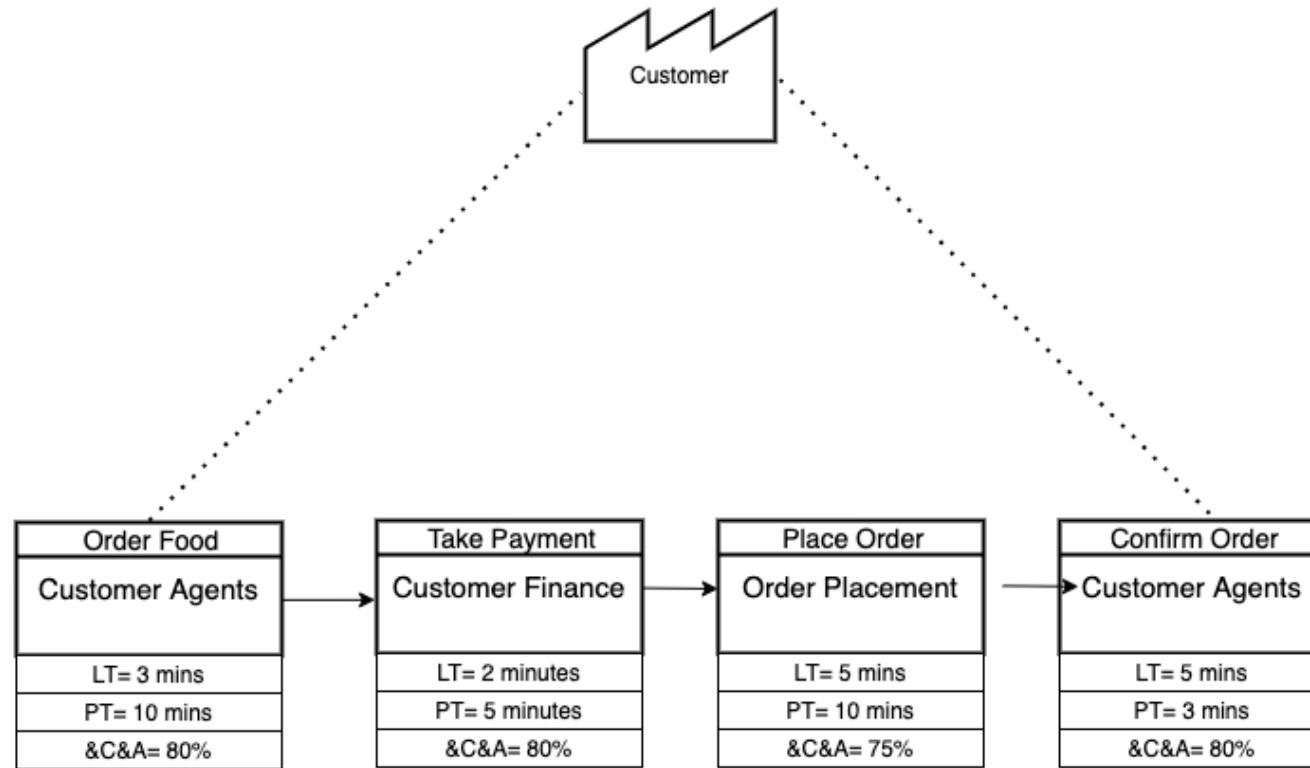
Upon Completion, Return to Ell Berman

Destroy

DPS/OE Form No. XXXXXXXXXX

Just Paper Takeaway

Current State Value Stream Map
Order Flow
Demand Rate 10/day
29 SEP 2021



1



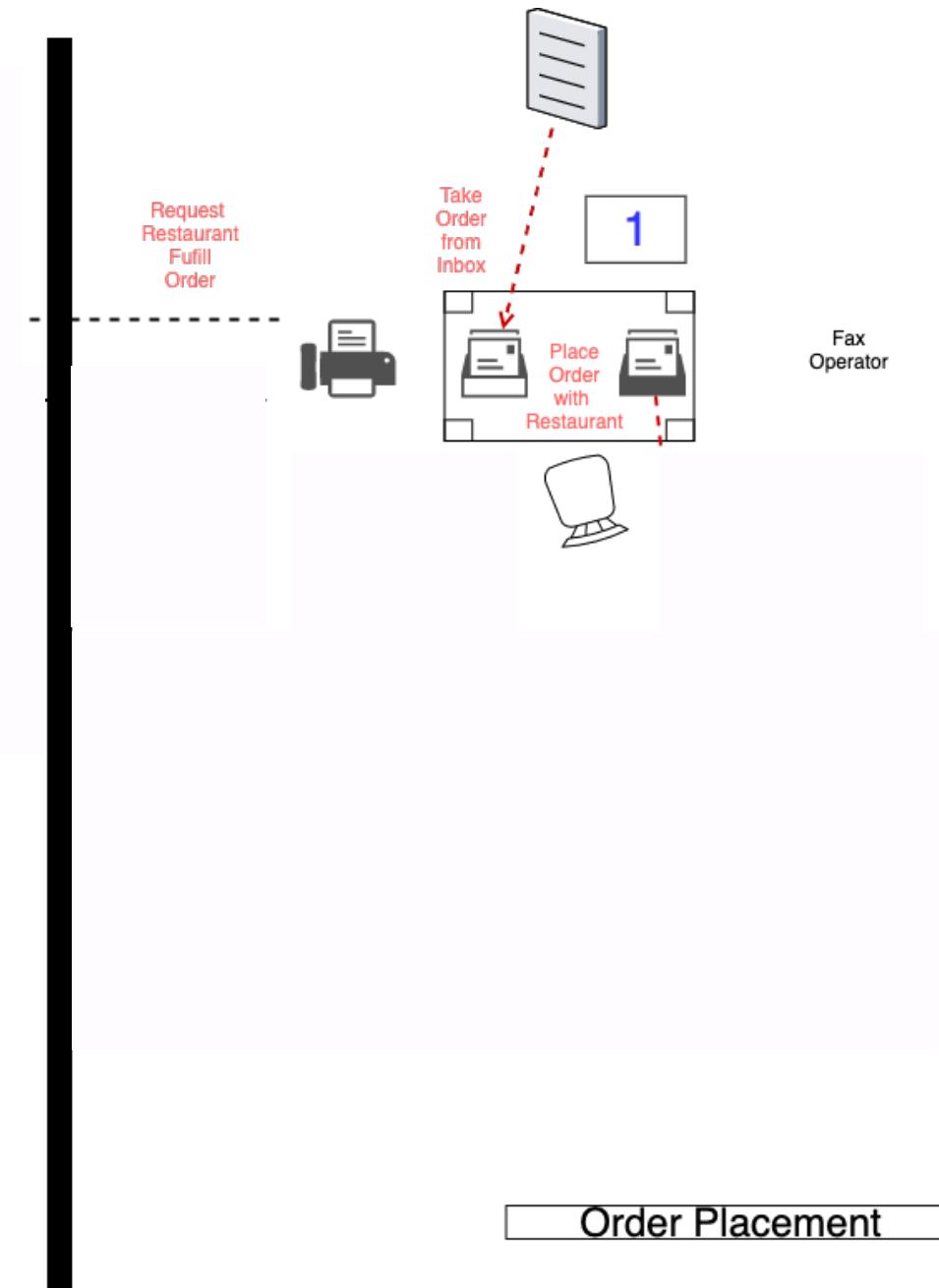
Hungry Customer

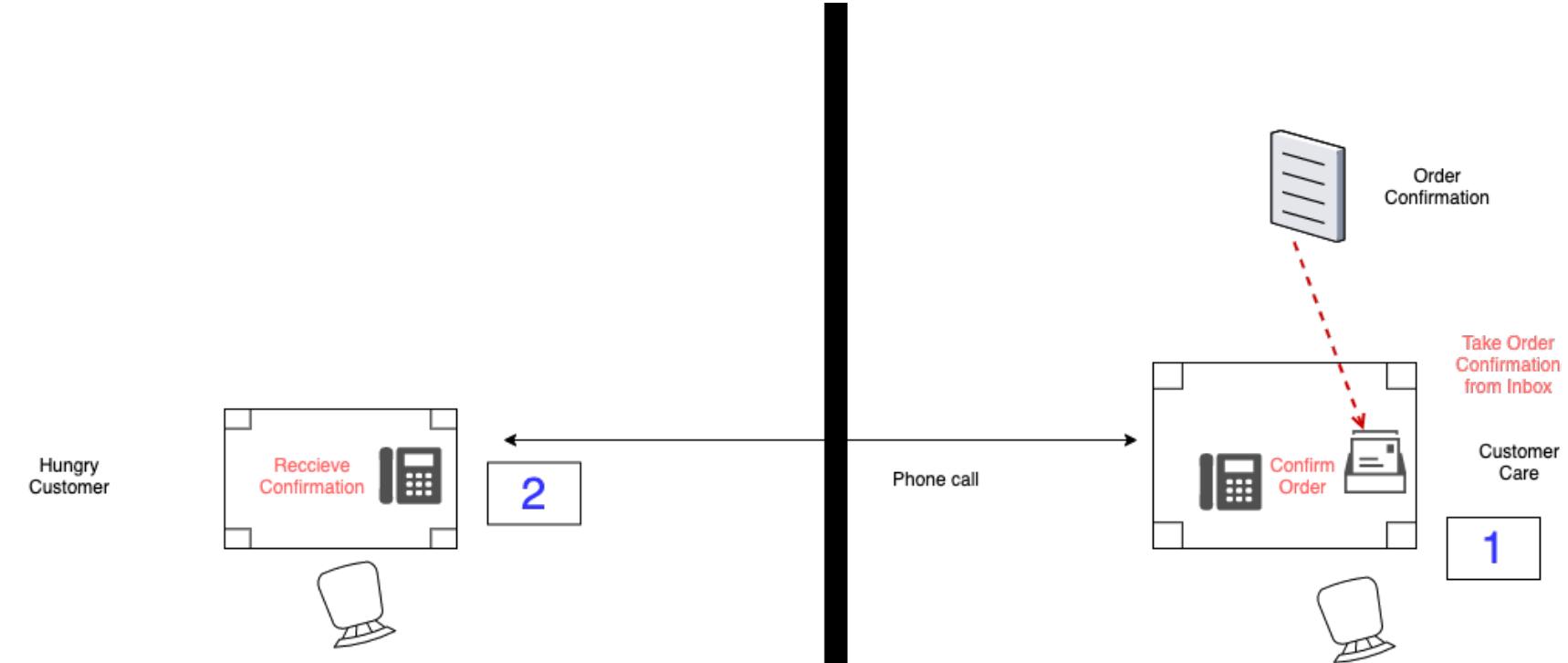


Order Taking

Order Wheel







Simplification

We are not going to show:

- The Order on the way to the customer
- The Order reaching the customer

Order Confirmation

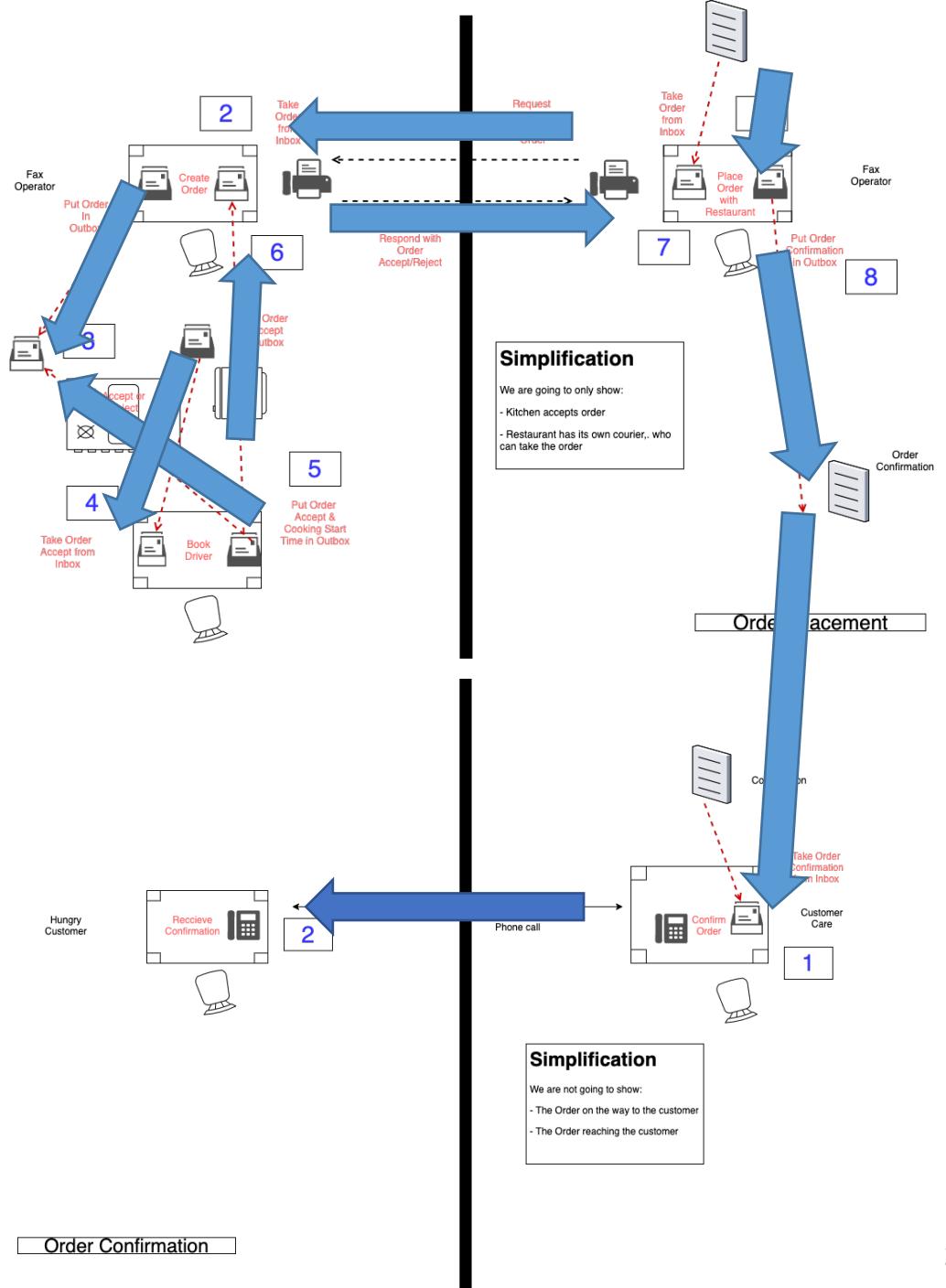
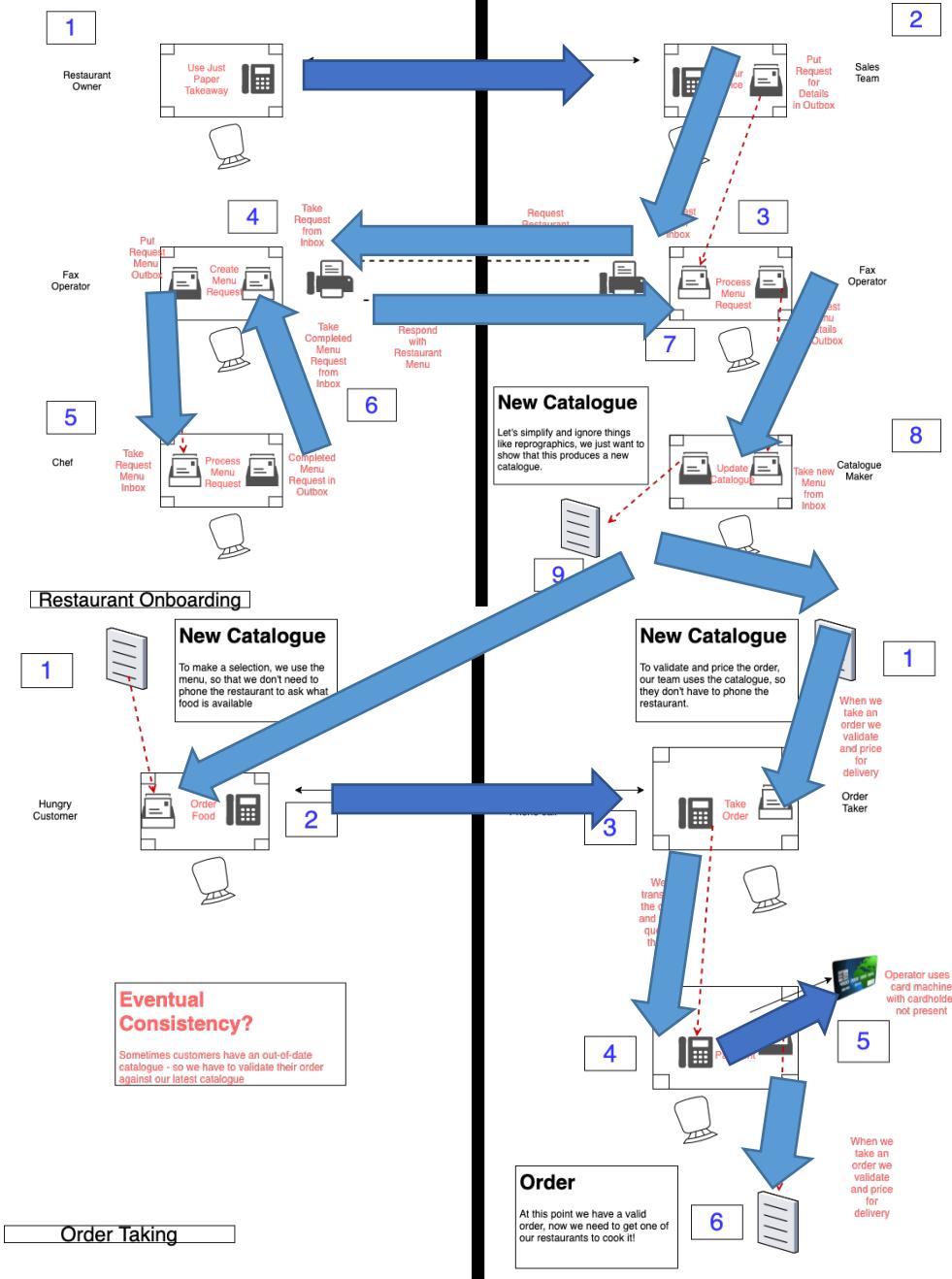
What Communication Style?



Flow: Asynchronous Conversation,
Information Packets moving along
Connectors between Components



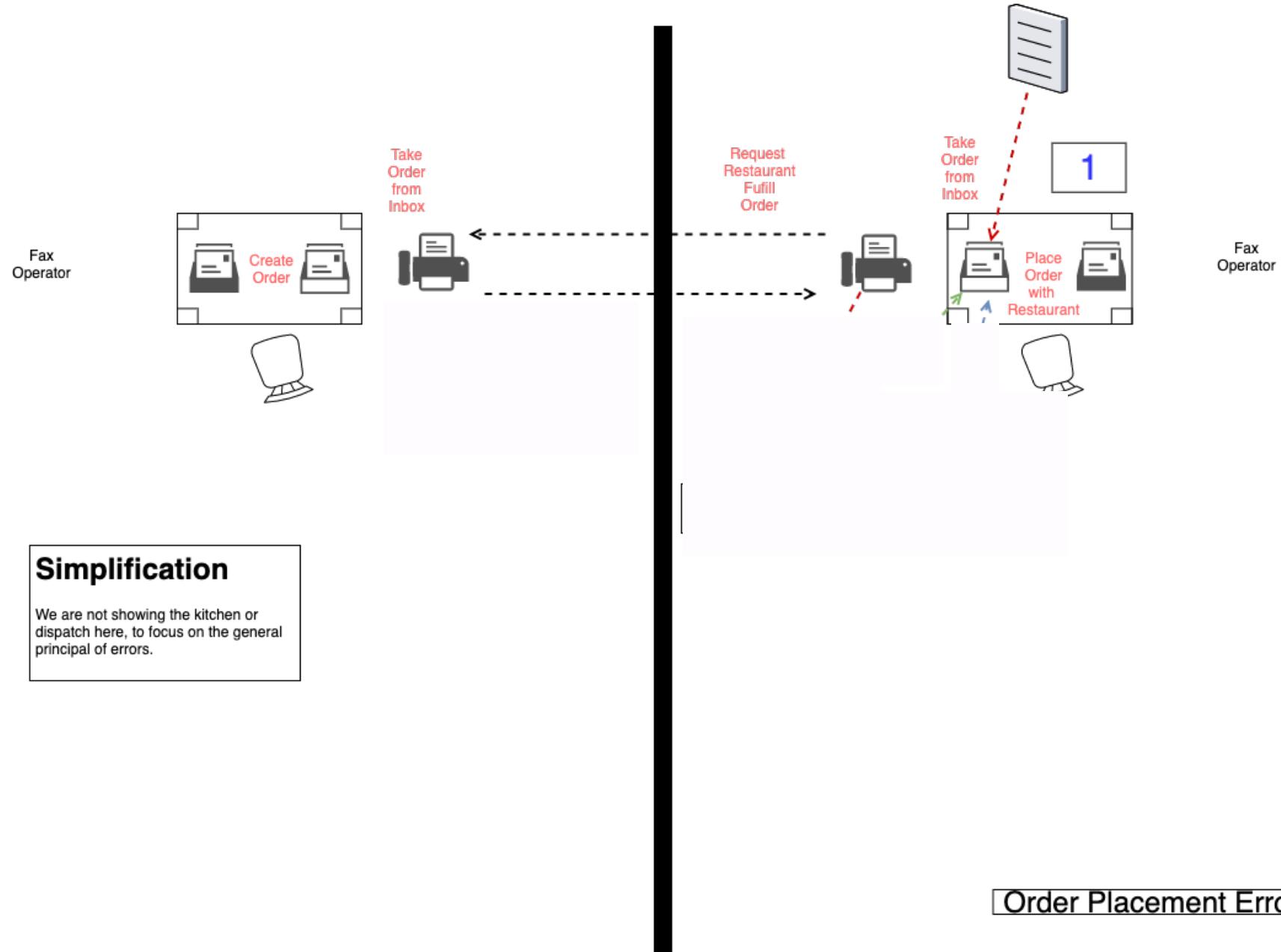
Hustle: Synchronous Conversation
interrupting when we are transforming one
Information Packet to another inside a
Component

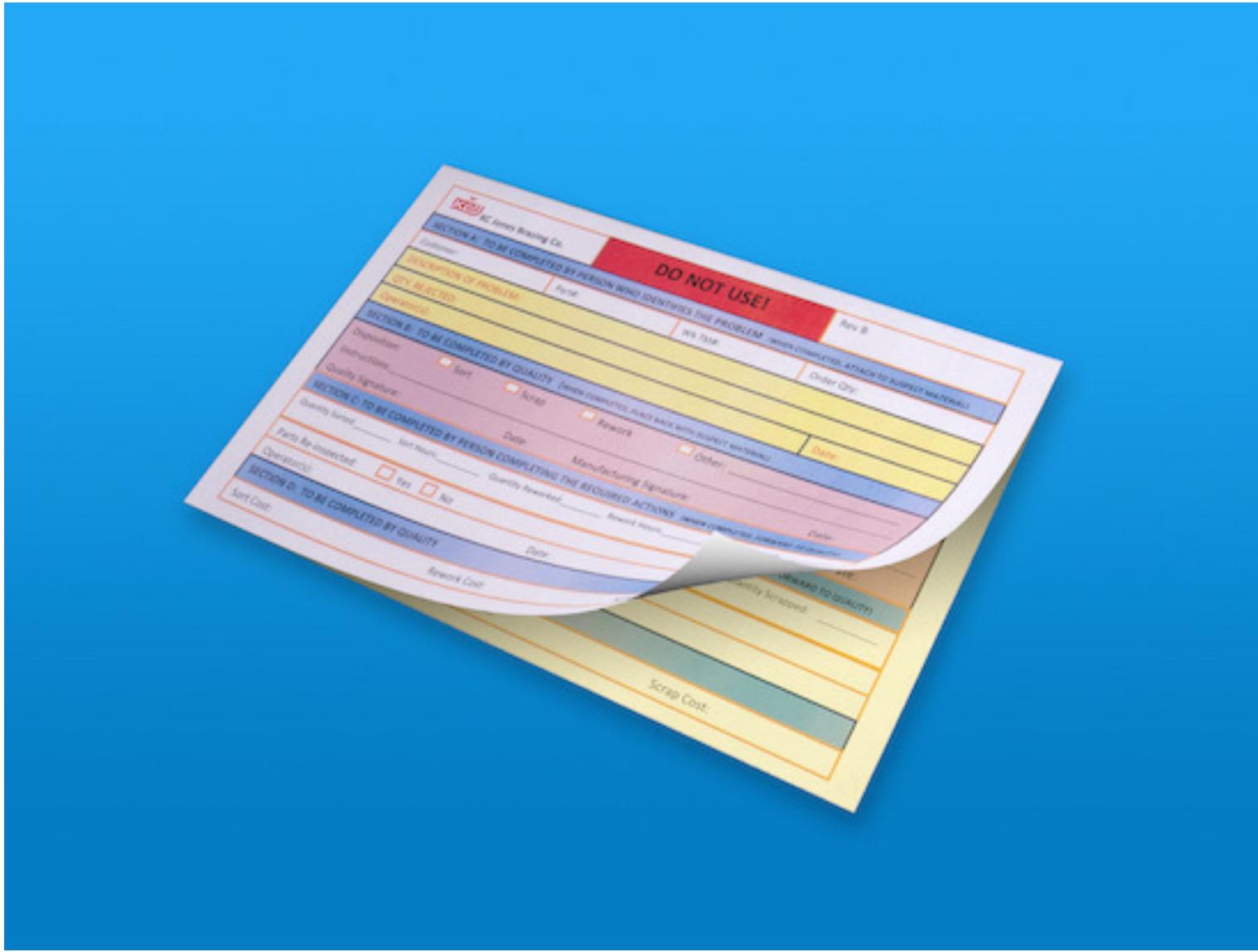


Fax Call Log

Monday, 2010-11-08 11:19

Date	Time	Type	Job #	Length	Speed	Station Name/Number	Pages	Status
2010-06-18	08:31	SCAN	92	0:25	28800	[REDACTED]	0	E-705 V.34 1M31
2010-03-22	11:39	RECV	59	0:20	26400	[REDACTED]	1	OK -- V.34 BM31
2010-03-22	11:45	RECV	60	0:20	26400	[REDACTED]	1	OK -- V.34 BM31
2010-03-22	12:29	RECV	61	0:21	26400	[REDACTED]	1	OK -- V.34 BM31
2010-09-14	14:46	SCAN	129	1:40	9600	[REDACTED]	2	E-606 V.29 AR30





1

Restaurant Owner



← Phone call →

Phone call

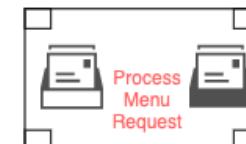
2

Sales Team



Simplification

We are not showing the kitchen or dispatch here, to focus on the general principal of errors.



Fax Operator



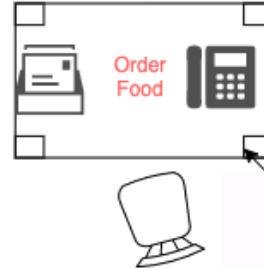
Restaurant Onboarding Errors

RECEIPT #: 14456
08-05-2015 10:21:12

TRANSACTION DECLINED

1

Hungry Customer

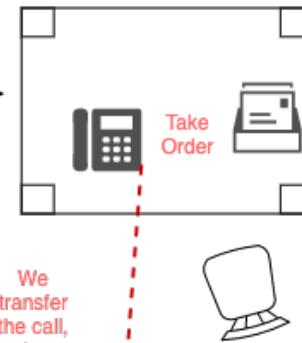


1

Phone call

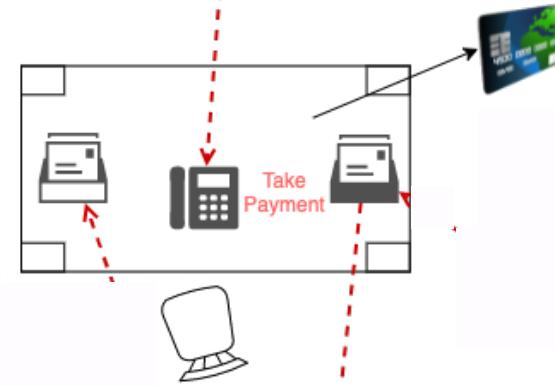
2

Order Taker



3

We transfer the call, and may queue them



Order Taking Errors

Flow Based Programming

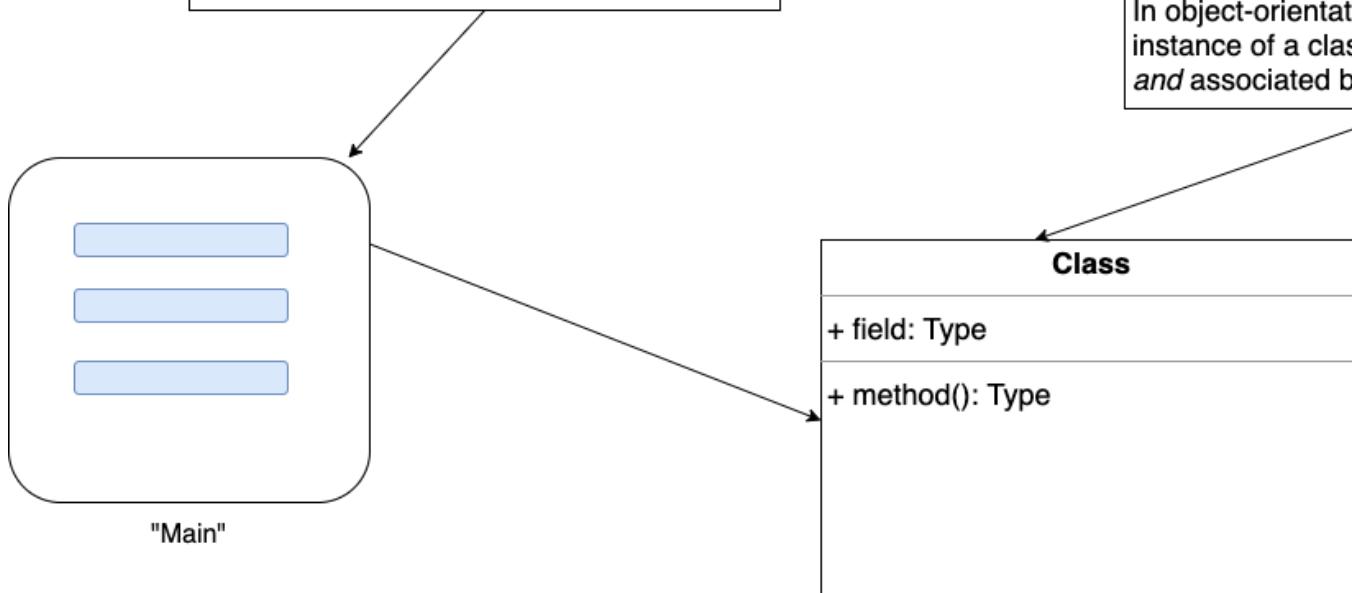
“Πάντα ῥεῖ – Everything flows” (Heraclitus of Ephesus, ca. 500 BCE)

J. Paul Morrison



Call and Return

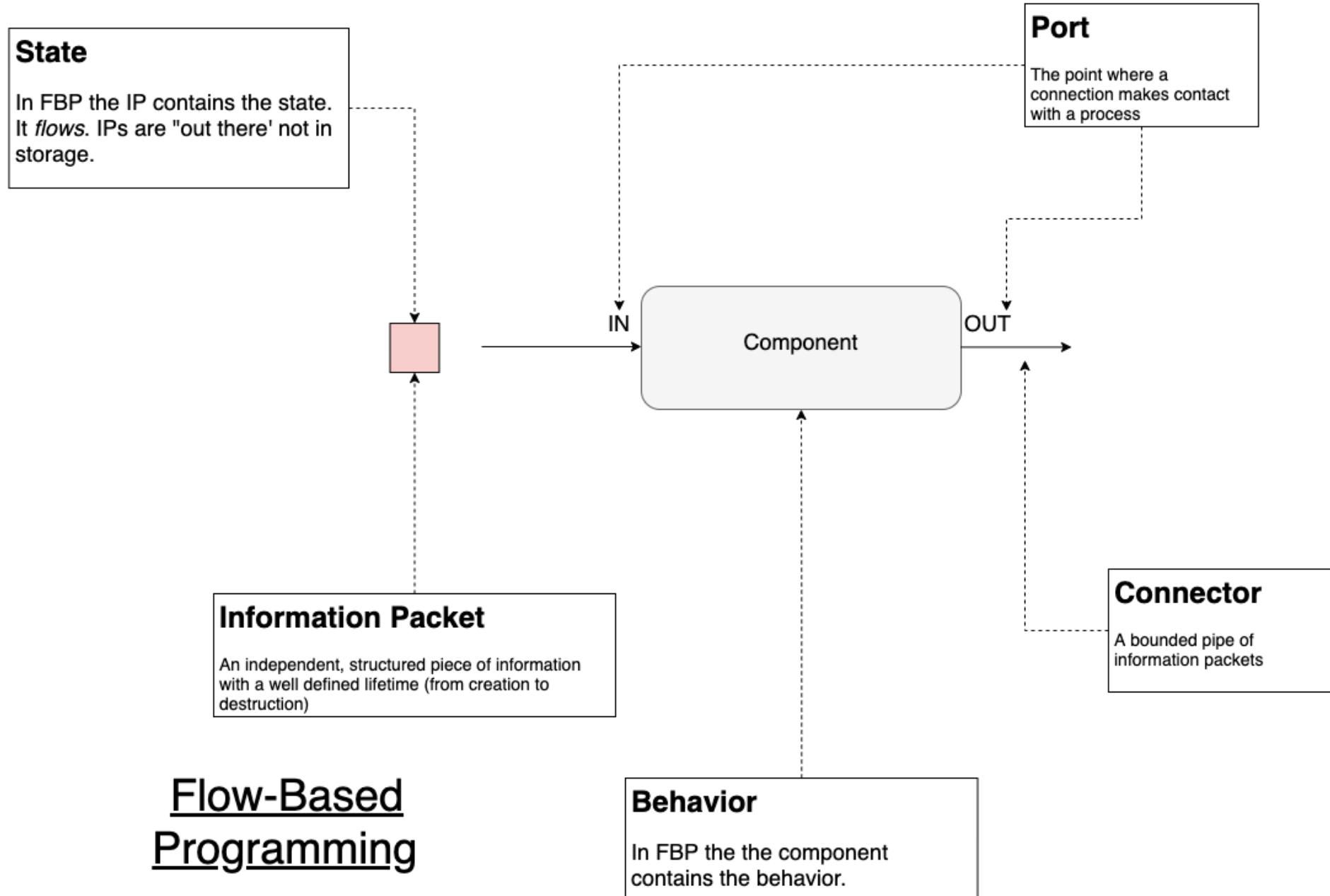
Flow of control is a "main" routine which calls out to "subroutines", in this case a method on a class, which returns control once done.



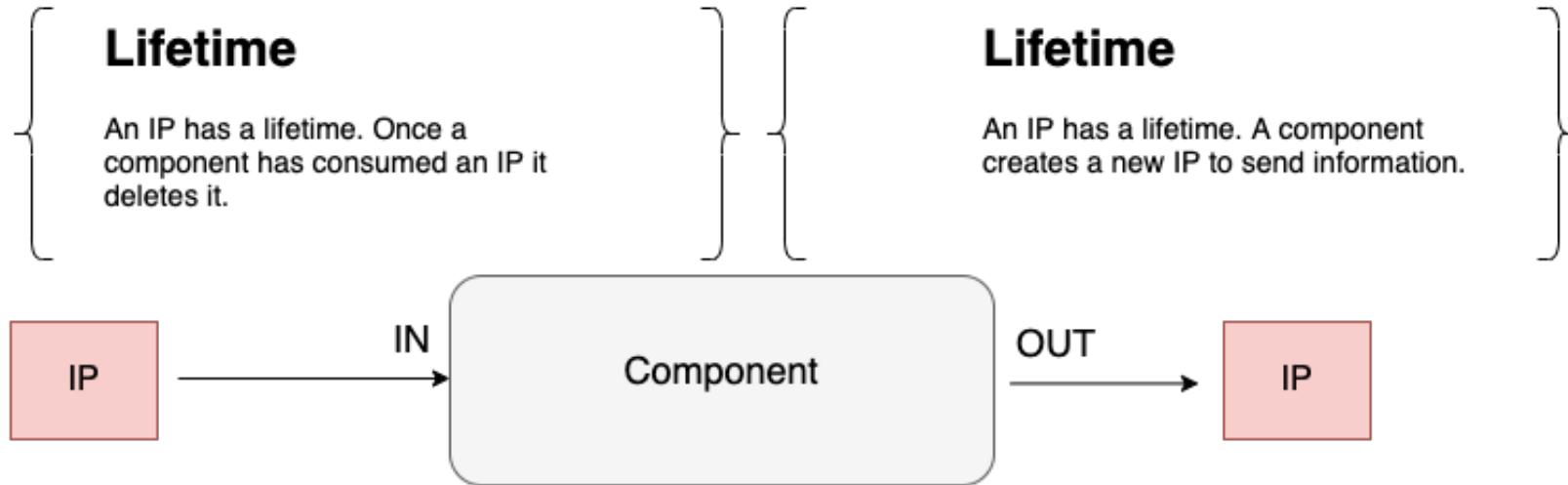
State and Behaviour

In object-orientation an object is an instance of a class that contains both state and associated behaviour.

Object Orientation



Flow-Based Programming



Flow-Based Programming

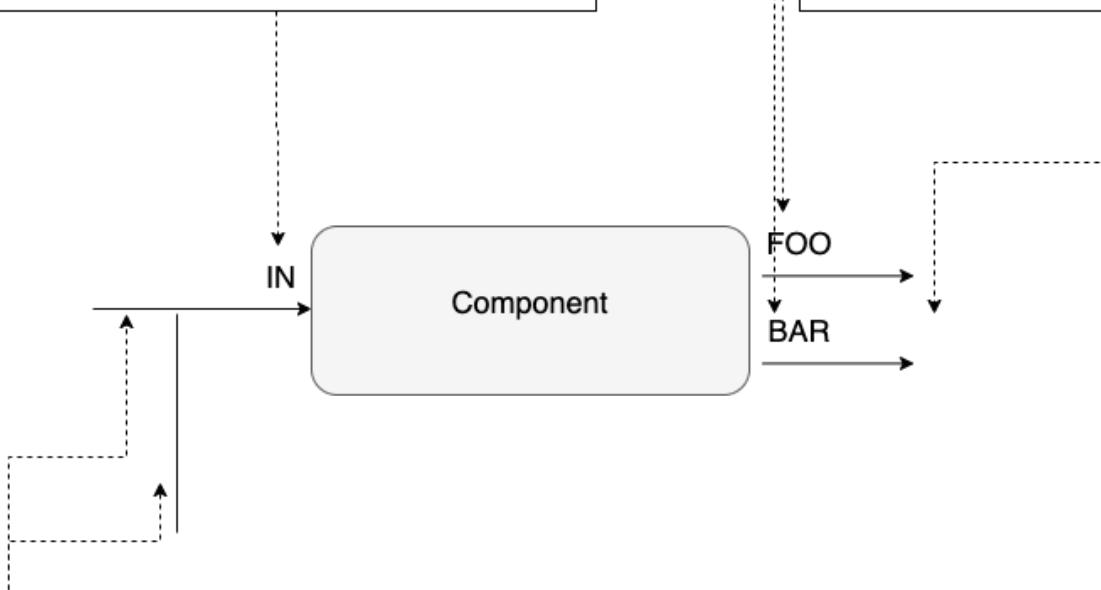
Ports and Connectors

Ports and Names

A port is abstracted away from the domain logic - it is a logical address from which things are read, or to which things are dispatched.

Multiple Ports and Connectors

We can have multiple ports and connectors, not just one



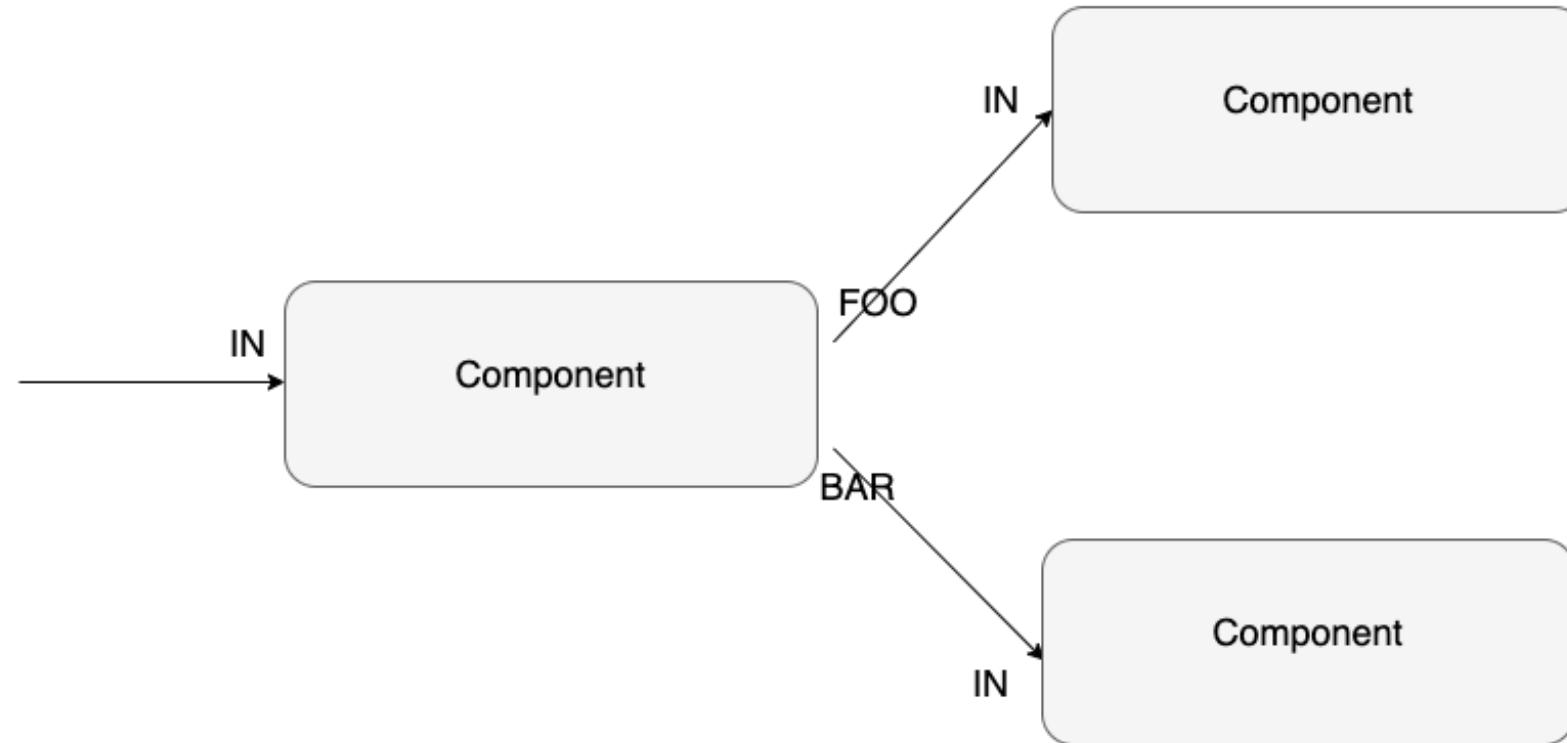
Multiple Writers to In

Multiple connections may arrive on a single IN port. This allows for sequencing.

Single Writers to Out

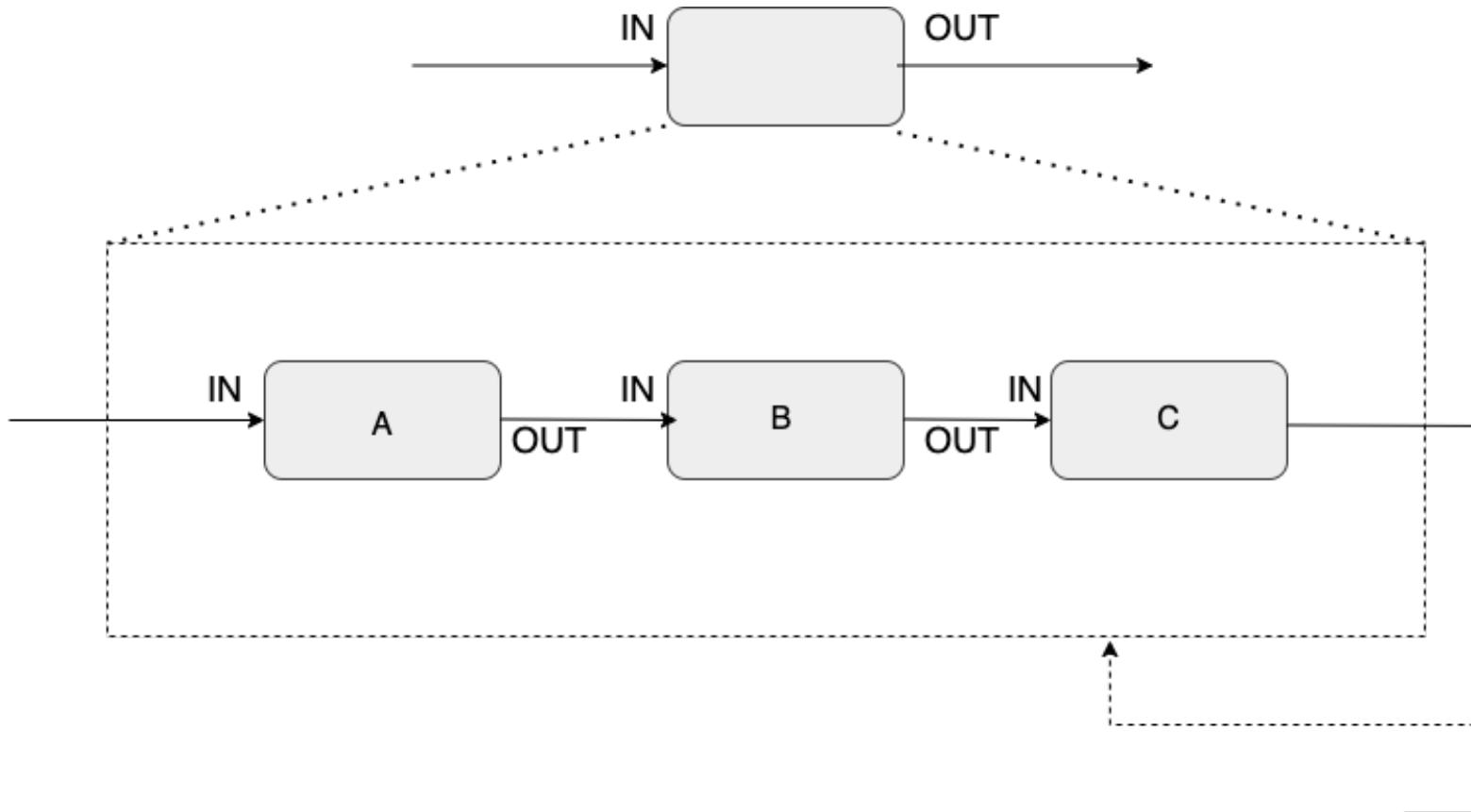
Each 'out' port only has a single connection

Networks



Flow

In FBP there we do not use "call and return" through a "main" routine, instead control flows through the application as IPs pass along connectors.

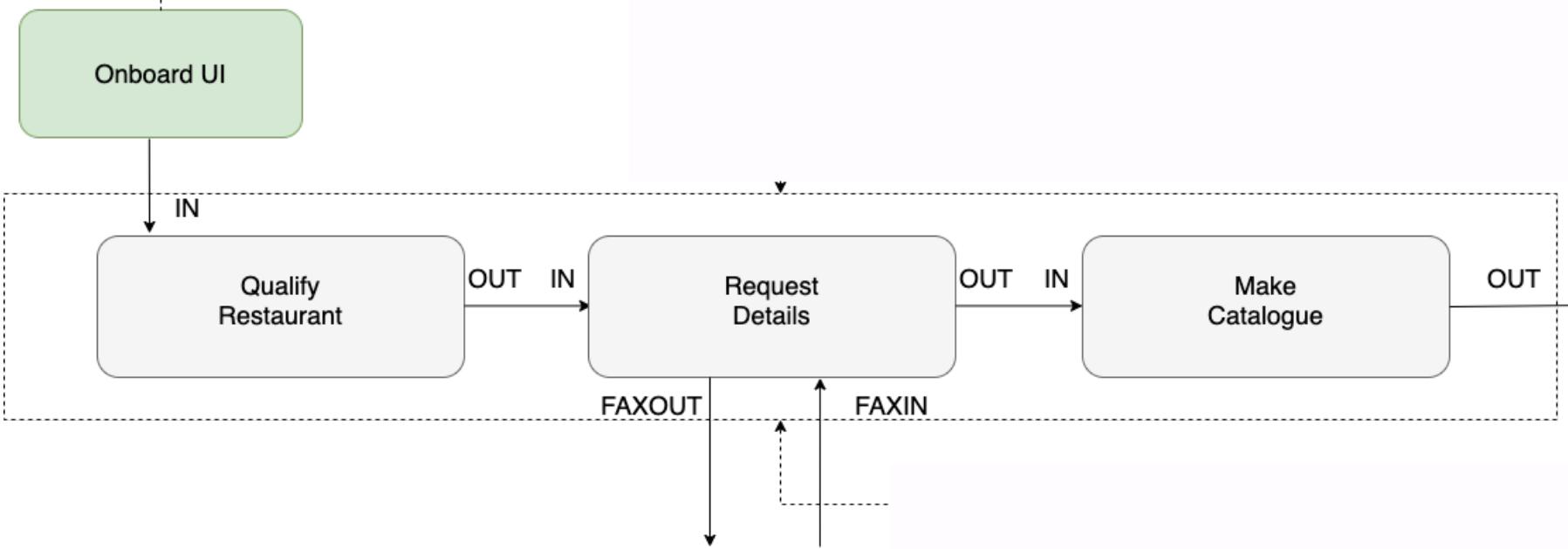


Composite Component

Subnet

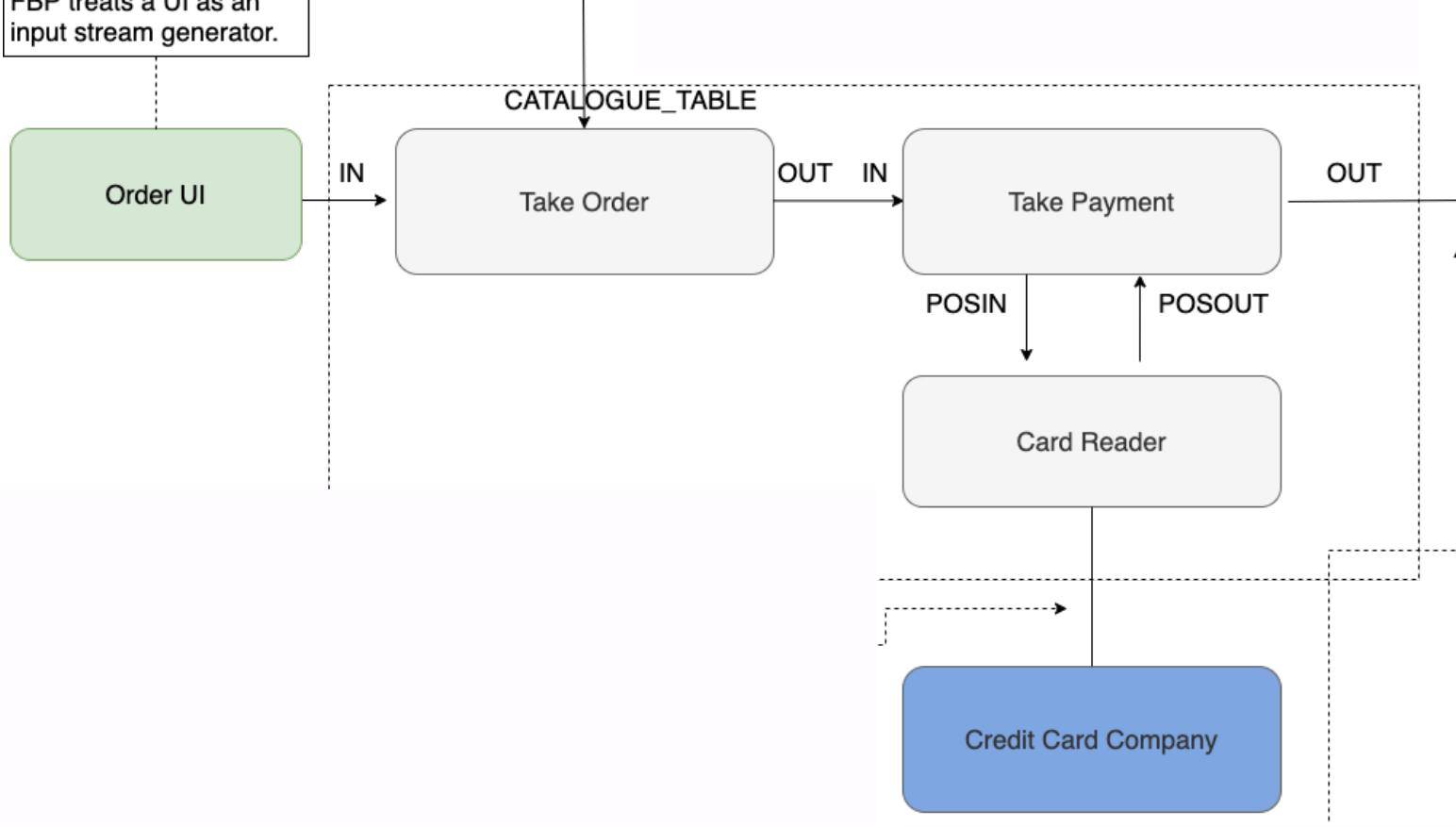
A network that forms a part of a larger network

UI
FBP treats a UI as an input stream generator.

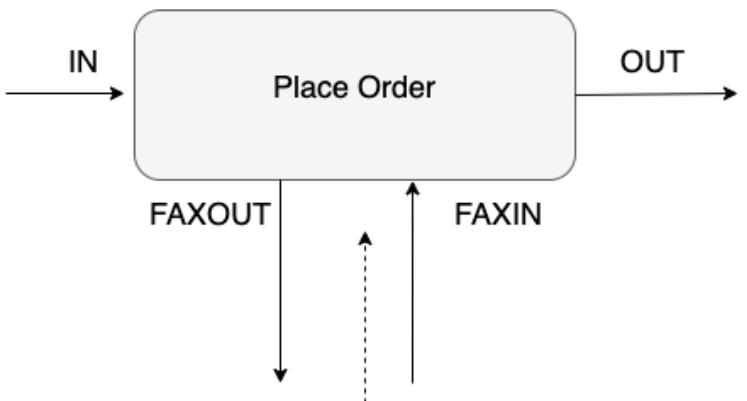


Onboard
Restaurant

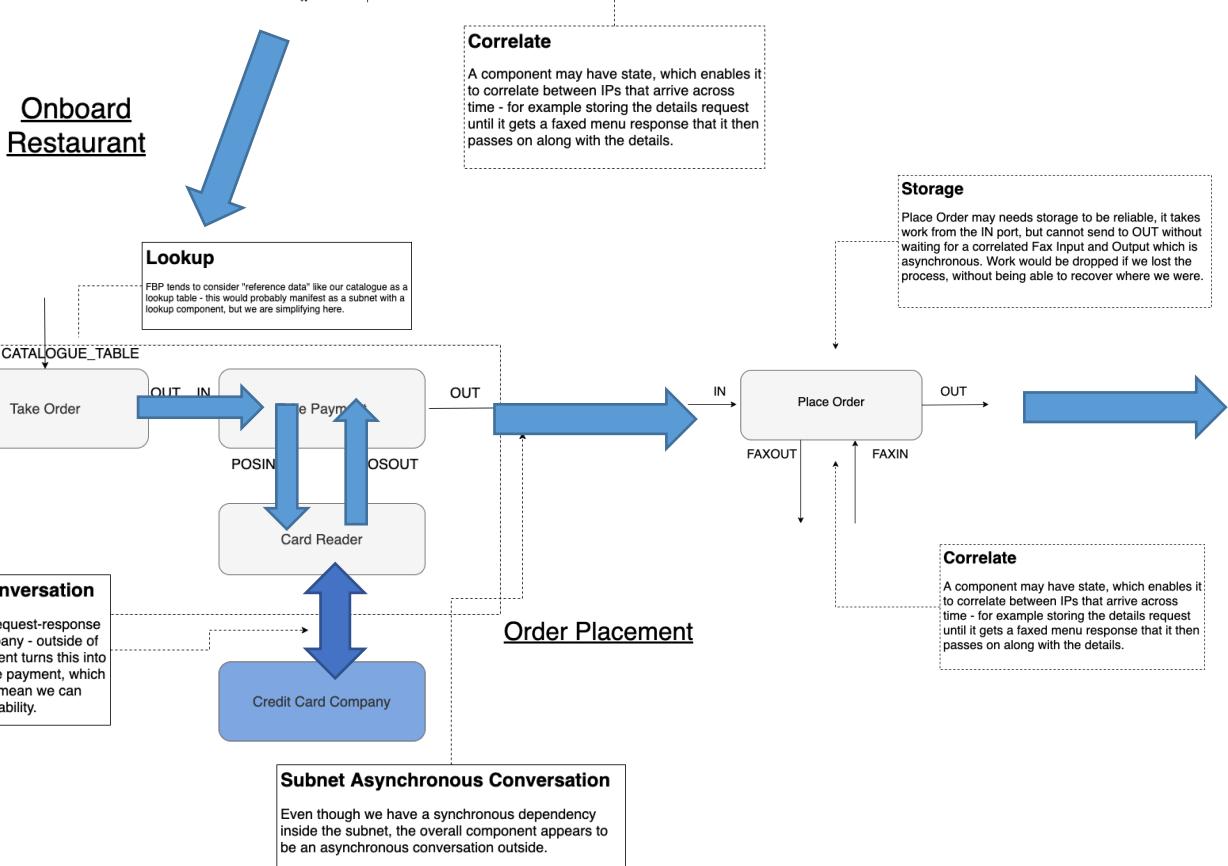
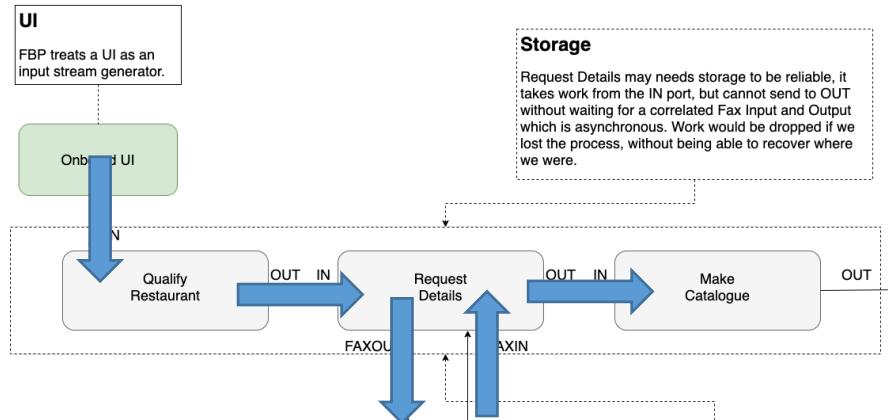
UI
FBP treats a UI as an input stream generator.



Order Food

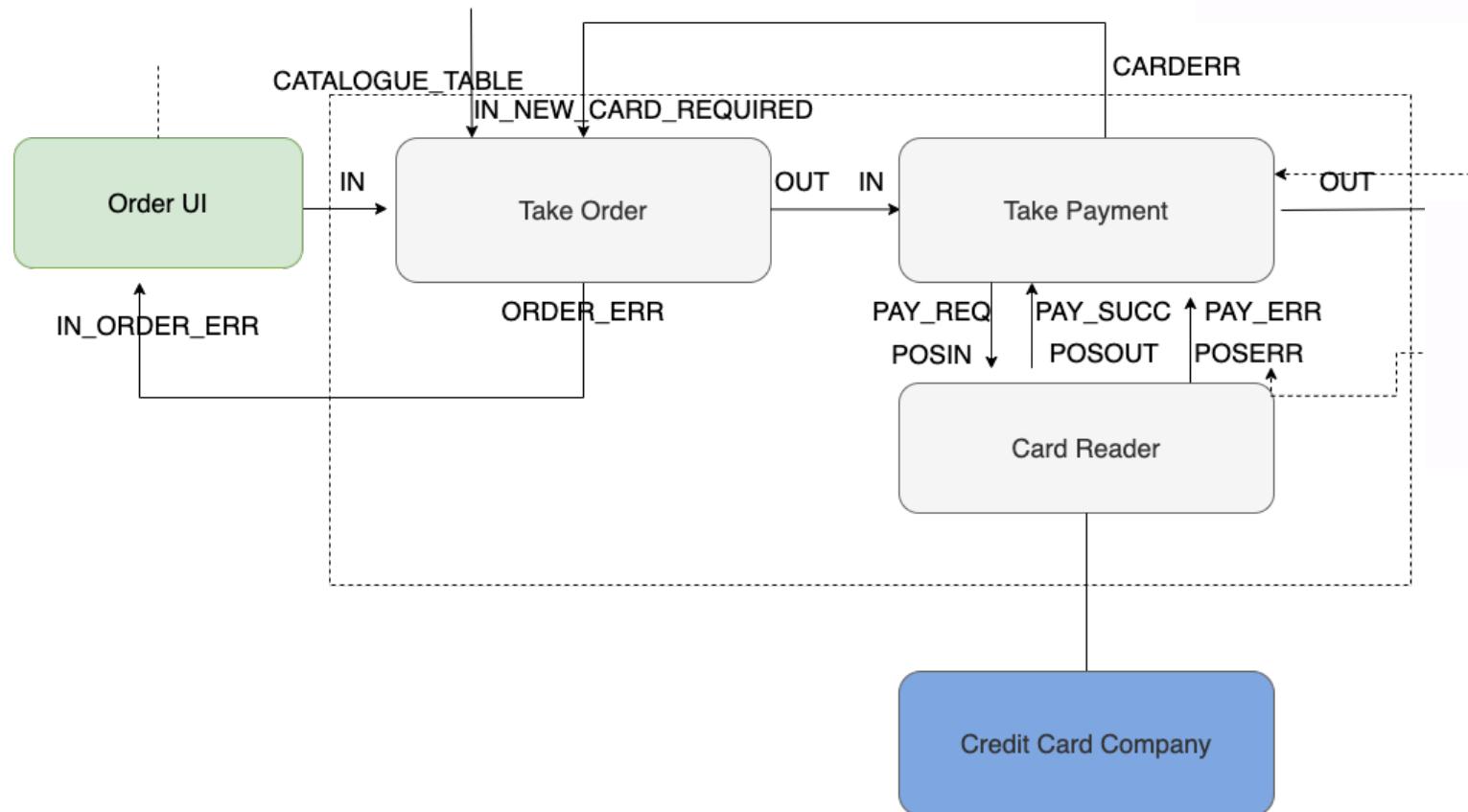


Order Placement



RECEIPT #: 14456
08-05-2015 10:21:12

TRANSACTION DECLINED



Order Food

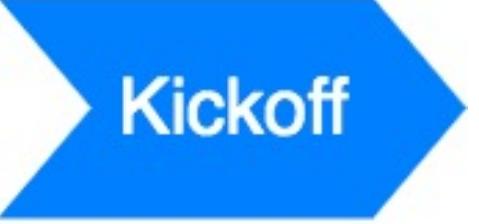
Hustle and Flow

“I’m in these heels every day and you never seen me fall down once!” – Nola, Hustle and Flow

Value Stream Mapping

A value stream is the sequence of activities required to design, produce, and deliver a good or service to a customer...

Martin, Karen. Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation.



Kickoff

Agree the scope to be mapped. We want to map how we create value for a customer i.e. Procure to Pay, or Order to Delivery; or we want to map how we support that, for example Analysis to Recommendation



Agree the scope to be mapped. We want to map how we create value for a customer i.e. Procure to Pay, or Order to Delivery; or we want to map how we support that, for example Analysis to Recommendation

Walk the value stream from end-to-end, identifying the processes.

As a rule, a new process is started when there is a delay between steps.

Look for 5-15 processes

Kickoff

Agree the scope to be mapped. We want to map how we create value for a customer i.e. Procure to Pay, or Order to Delivery; or we want to map how we support that, for example Analysis to Recommendation

First Walk

Walk the value stream from end-to-end, identifying the processes.

As a rule, a new process is started when there is a delay between steps.

Look for 5-15 processes

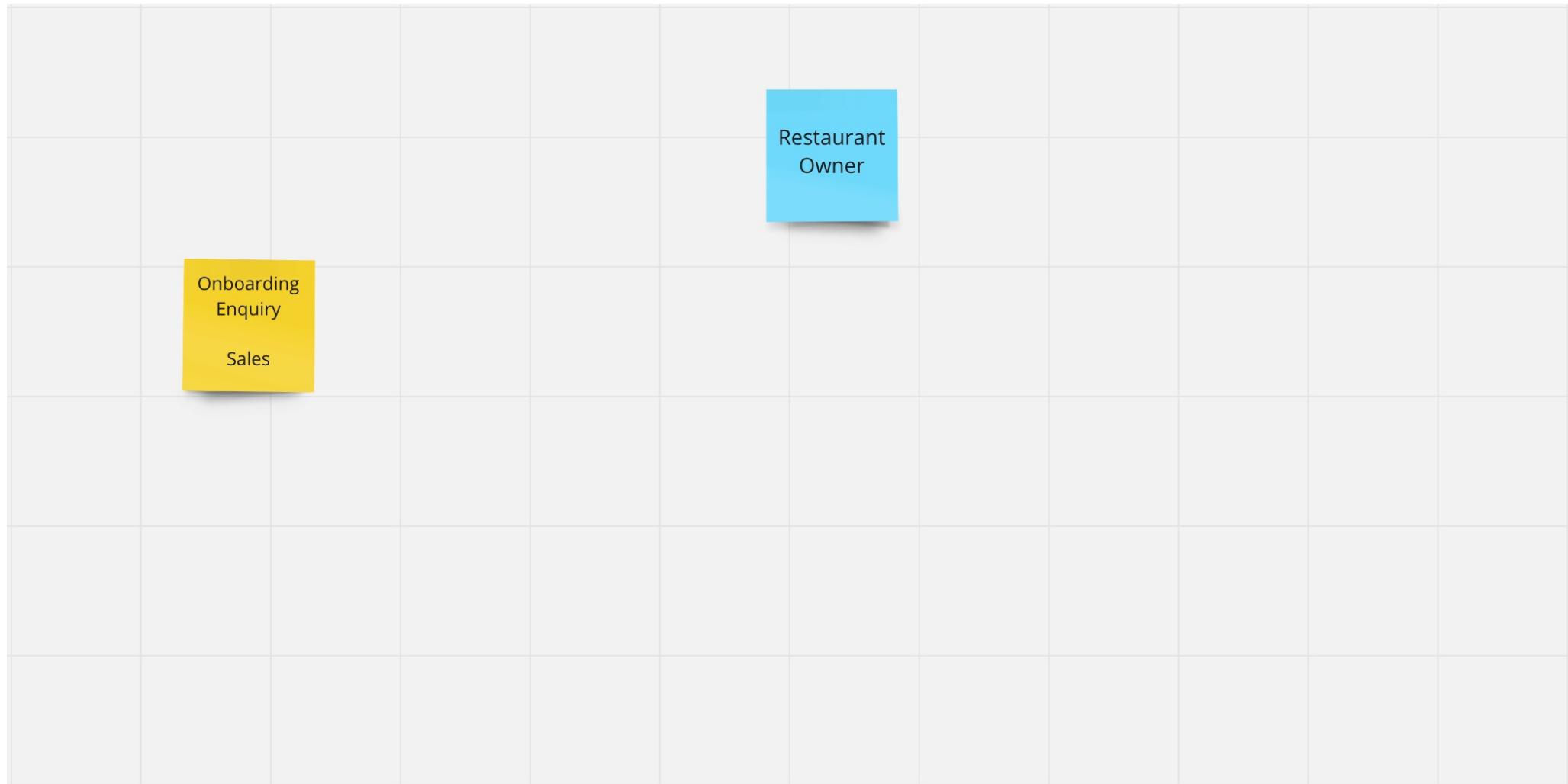
Map Layout

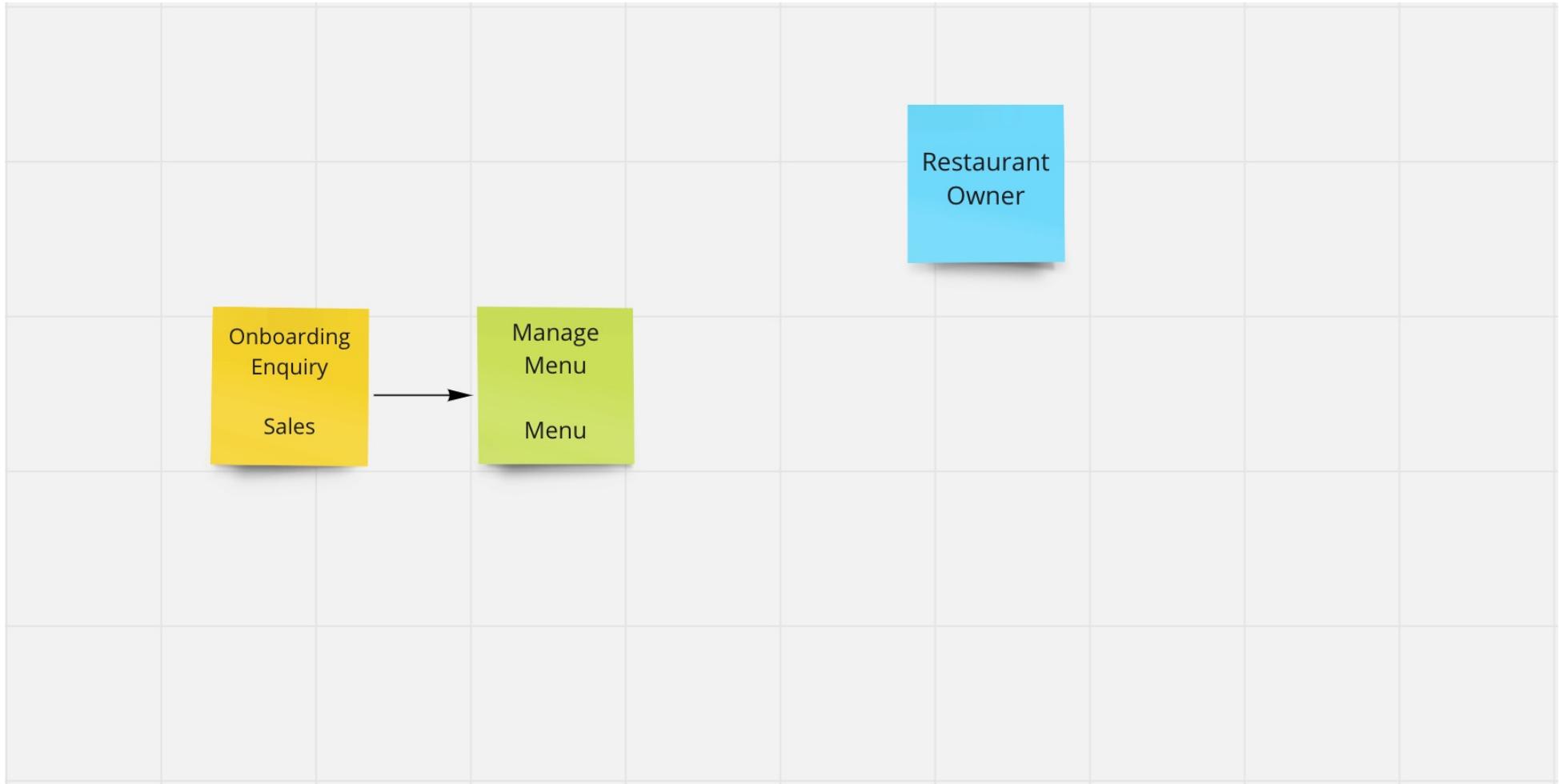
Place a customer Post-It note at the top.

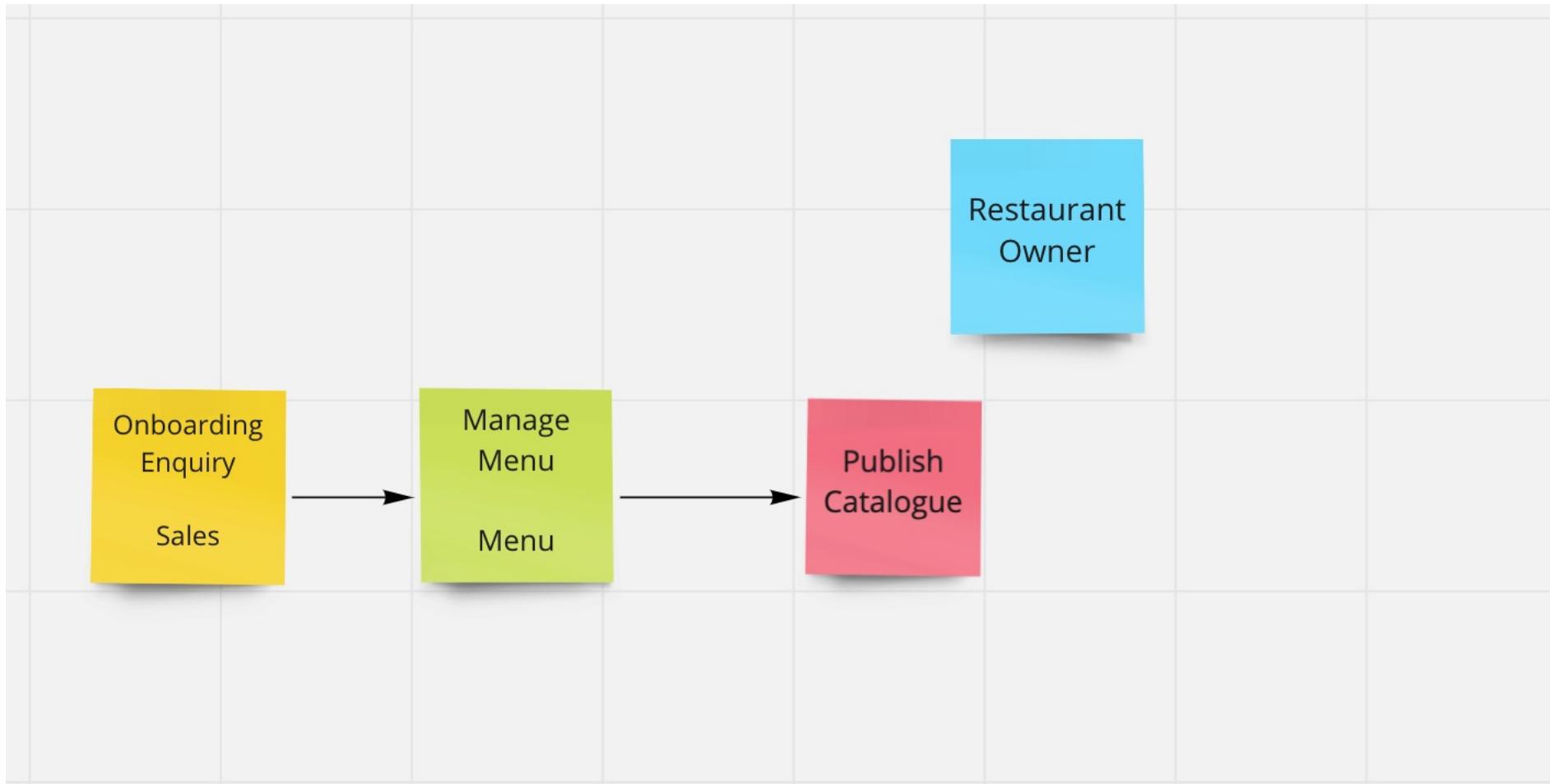
Below that, horizontally, and left-to-right add Post-Its to represent the processes in the value stream.

Avoid forks, map the most common path to create value for the customer.

Number the process blocks.







Kickoff

Agree the scope to be mapped. We want to map how we create value for a customer i.e. Procure to Pay, or Order to Delivery; or we want to map how we support that, for example Analysis to Recommendation

1st Walk

Walk the value stream from end-to-end, identifying the processes.

As a rule, a new process is started when there is a delay between steps.

Look for 5-15 processes

Map Layout

Place a customer Post-It note at the top.

Below that, horizontally, and left-to-right add Post-Its to represent the processes in the value stream.

Avoid forks, map the most common path to create value for the customer.

Number the process blocks.

2nd Walk

Walk the value stream from backwards, looking for processes that the initial walk through missed..

Track key metrics (Process Time, Lead Time, % complete and accurate)

Hold a census of the number of people

Kickoff

Agree the scope to be mapped. We want to map how we create value for a customer i.e. Procure to Pay, or Order to Delivery; or we want to map how we support that, for example Analysis to Recommendation

1st Walk

Walk the value stream from end-to-end, identifying the processes.

As a rule, a new process is started when there is a delay between steps.

Look for 5-15 processes

Map Layout

Place a customer Post-It note at the top.

Below that, horizontally, and left-to-right add Post-Its to represent the processes in the value stream.

Avoid forks, map the most common path to create value for the customer.

Number the process blocks.

2nd Walk

Walk the value stream from backwards, looking for processes that the initial walk through missed..

Track key metrics (Process Time, Lead Time, % complete and accurate)

Hold a census of the number of people

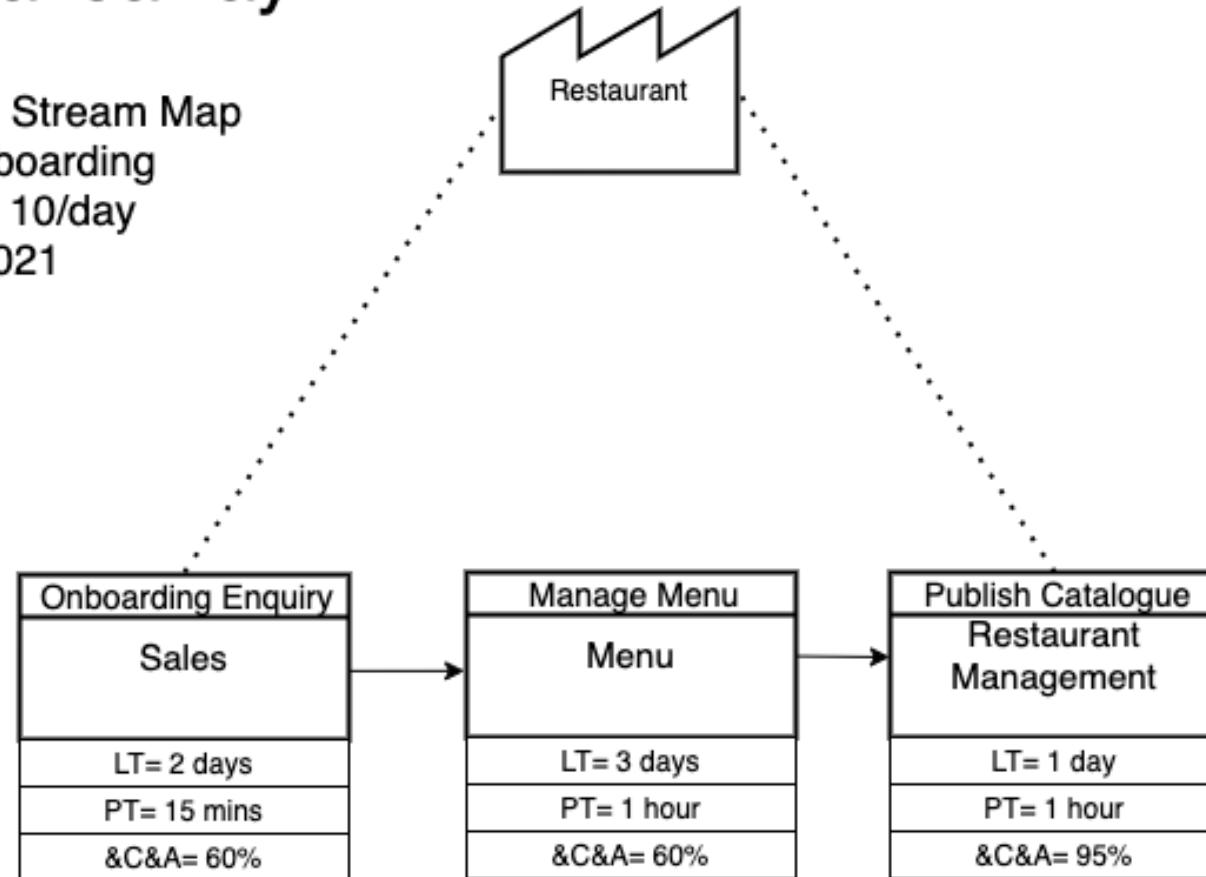
Map Details

Add the details from the second walk to the map.

Add A timeline and map summary to the bottom axis of the map

Just Paper Takeaway

Current State Value Stream Map
Restaurant Onboarding
Demand Rate 10/day
29 SEP 2021



Event Storming

EventStorming is a workshop format for quickly exploring complex business domains.

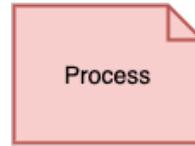
Alberto Brandolini - <https://ziobrando.blogspot.com/2013/11/introducing-event-storming.html>

Event Storming

- It focuses on identifying the Information Packets – called events in this case – that pass through the system
- Once the events have been identified, the ‘gaps’ that transform one event to another are identified as process or aggregates.
- Commands initiate flows, and view models help us identify how commands are created.



An information packet that flows, something placed in an Outbox for transmission to interested parties
Inboxes



Responds to and transforms an information packet



State that is updated. Event Storming uses these a lot, due to DDD heritage, from FBP insights: they are less important than a process, as state is less important than flow.



A read-only model built from events (information packets) that we can use to make decisions i.e. commands



Initiates flow

Event Storming - Key

Phase One - Capture Events

In phase one, you capture events, you can think of events as the IPs flowing through the system or as what goes into an Outbox.



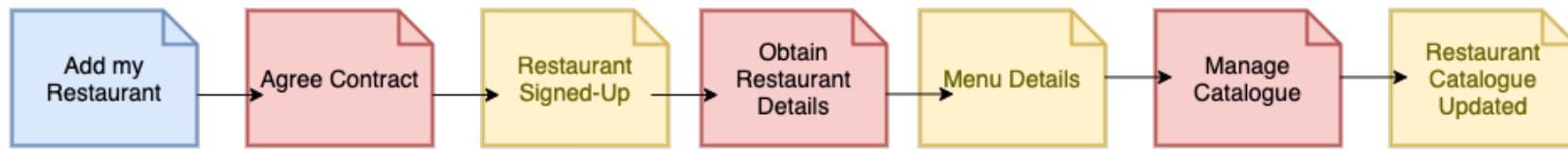
Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Restaurant Onboarding

Phase Two - Commands, Processes, Aggregates

In phase 2, the processes/aggregates are the 'hustle' work at the desk that transforms inbox to outbox



Event Storming - Restaurant Onboarding

Phase One - Capture Events

In phase one, you capture events, you can think of events as the IPs flowing through the system or as what goes into an Outbox.



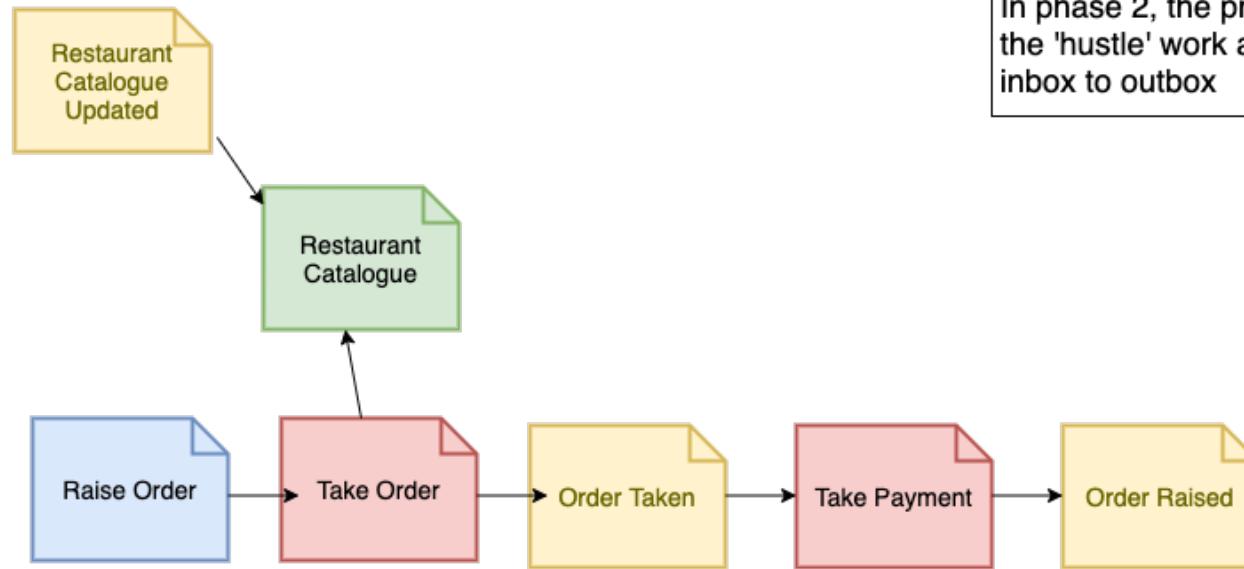
Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Customer Order

Phase Two - Commands, Processes, Aggregates

In phase 2, the processes/aggregates are the 'hustle' work at the desk that transforms inbox to outbox



Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Customer Order

Phase One - Capture Events

In phase one, you capture events, you can think of events as the IPs flowing through the system or as what goes into an Outbox.



Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Order Placement

Phase Two - Commands, Processes, Aggregates

In phase 2, the processes/aggregates are the 'hustle' work at the desk that transforms inbox to outbox



Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Order Placement

Phase One - Capture Events

In phase one, you capture events, you can think of events as the IPs flowing through the system or as what goes into an Outbox.



Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Order Confirmation

Phase Two - Commands, Processes, Aggregates

In phase 2, the processes/aggregates are the 'hustle' work at the desk that transforms inbox to outbox



Simplification

We only show the Just Paper Takeaway side here, not the restaurant, which is not under our control and we assume the interaction with the fax is "off stage".

Event Storming - Order Confirmation

Event Modelling

- Event Modelling, created by Adam Dymitruk is a technique for modelling flow
- It focuses on identifying the Information Packets – called events in this case – that pass through the system
- It models in more detail, the information needed to make commands that initiate flows,
- For any command that needs to use information – making a menu choice for example – there needs to be a view, built from events that provide the information required.

T:05

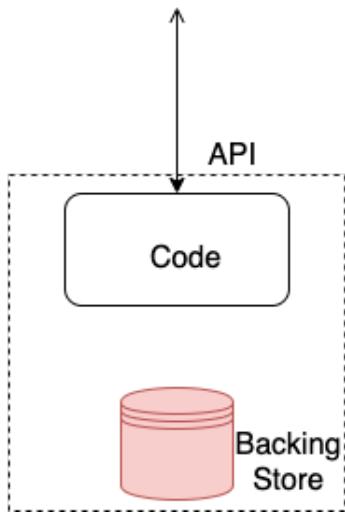


SERVICES & FLOW

Services and Flow

Web Service (SOA era)

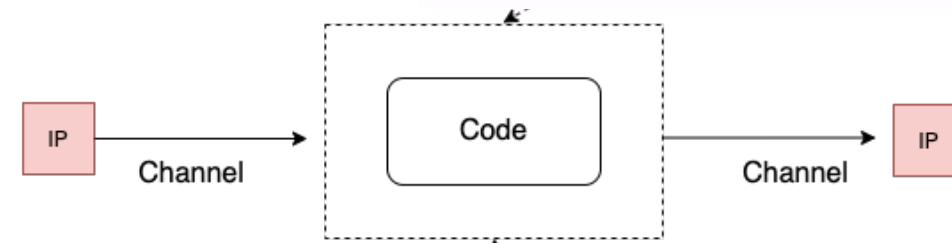
The service holds both state and behavior. State is an *implementation detail* of the service. Access to state is via an API



Services and Flow

FBP Service (Microservices era)

The service has behavior. State lives in packets of information in *channels* between services. Access to state is via consumption from a *channel*.



Q&A