

# Storming Our Way to Success

By Kristi Salmi

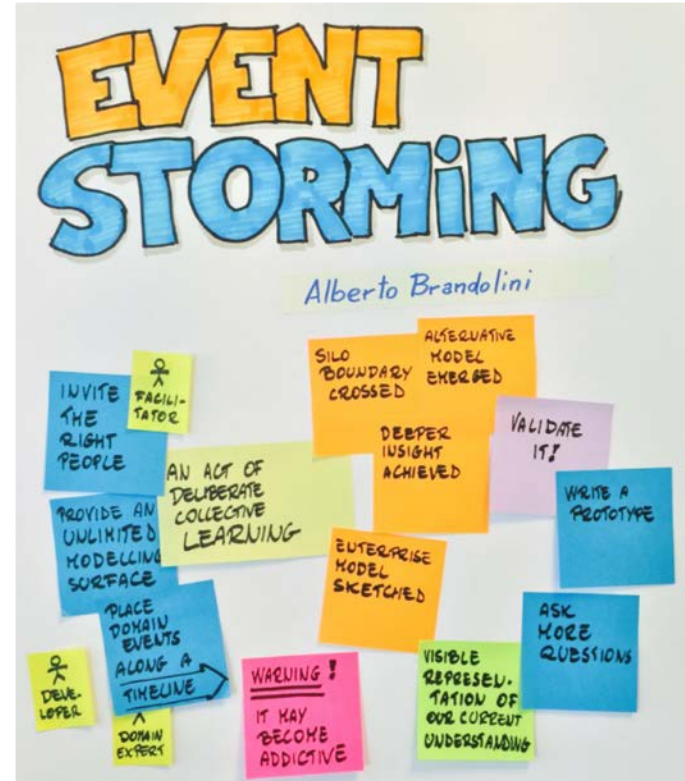
A dark blue diagonal graphic that starts from the bottom left corner and extends towards the top right corner, creating a triangular shape in the bottom right of the page.

# Introduction

- About me:
  - I'm Kristi
  - Background in data analytics within the financial services industry
  - Now Head of Product and Data with a RegTech company
- Topic & Case-Study for today:
  - Using Event Storming for the New Payments Platform (NPP) project
  - Based on a previous role with a Bank

# Event Storming – What is it?

- Event Storming origins:
  - DDD & Gamestorming
- Created by Alberto Brandolini
- Fun, lightweight & collaborative to quickly discover & explore:
  - Business processes
  - Related logical architecture
- Interactive session to grasp the process using
  - White board / wall with butchers paper
  - Lots of post-it notes
- Starts out:
  - Mapping business process
  - Before including technical aspects
- Used for:
  - Large & complex problems
  - Also smaller, simpler ones



# Event Storming – Why is it Successful?

- What makes Event Storming so successful?
  - It brings the entire team (Business & IT) together
  - Establishes a common understanding of:
    - Terminology
    - The process
    - The end goal & value/outcome to achieve
  - End goal is providing value to users
  - Common / ubiquitous language



# Benefits & Why I Like It

- Benefits I've found:
  - Collaboration & communication
  - Shared understanding
  - Based on Empiricism
    - Caters for unknowns
    - Make changes later
  - Easier & cheaper to move sticky-notes than writing code
- Why I like the process
  - First time for loan origination process
  - 1 day we'd mapped it out
  - Everyone knew what needed to be done
  - All on the same page
  - I thought "why don't we always do this"?
  - Removes need for heavy documentation spec's



# Key Outcomes Achieved

- Within the 12 weeks team was able to deliver:
  - Working software
  - Register a PayID
  - Receive inbound payments
- Achieved more in the 12 weeks than in the previous 2+ years
- Largely due to changes in ways of working
  - Including adopting Event Storming
- Technique I've instantiated at:
  - My organisation for client engagements
  - With another neo-bank client



# Key Take-Aways for Today

- Key take-aways for today:
  - Key learnings to run a successful session
  - Overview of the process
  - How you can do this today



# Challenge – NPP

- Working on implementing NPP - New Payments Platform
  - Near real-time & real-time payments
  - Choose PayID -> Mobile Number or Email for payments
- Project been running for 2 years
  - Nothing substantial developed
  - Behind & not going to deliver on time
  - Other FIs rolling-out NPP
- Decision made to:
  - Run NPP in innovation lab for 12 weeks
  - In-conjunction with Red Hat
  - Made changes to:
    - Way we worked
      - Event Storming -> requirements & solution design
    - Tools and technology used

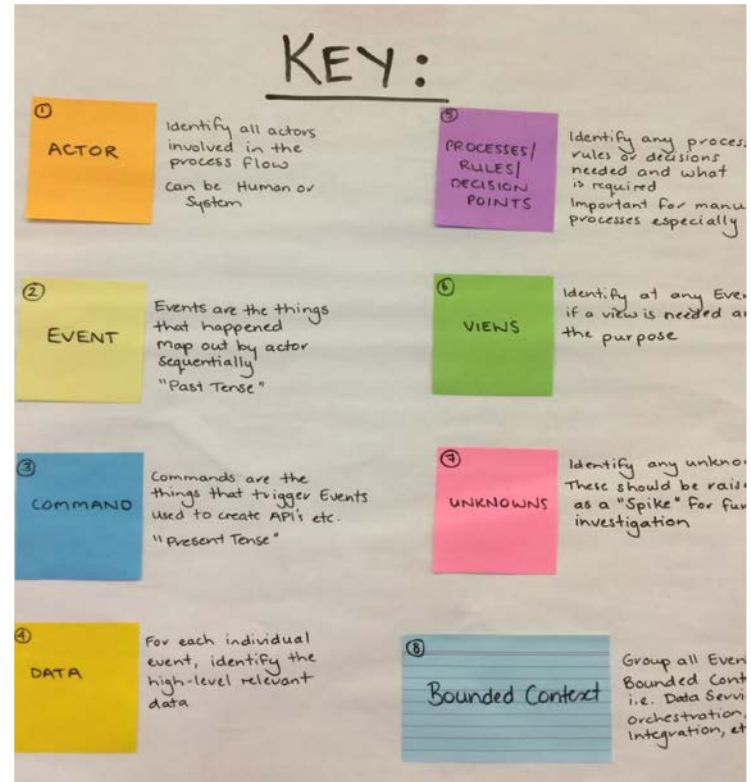




# Event Storming – Overview of Process

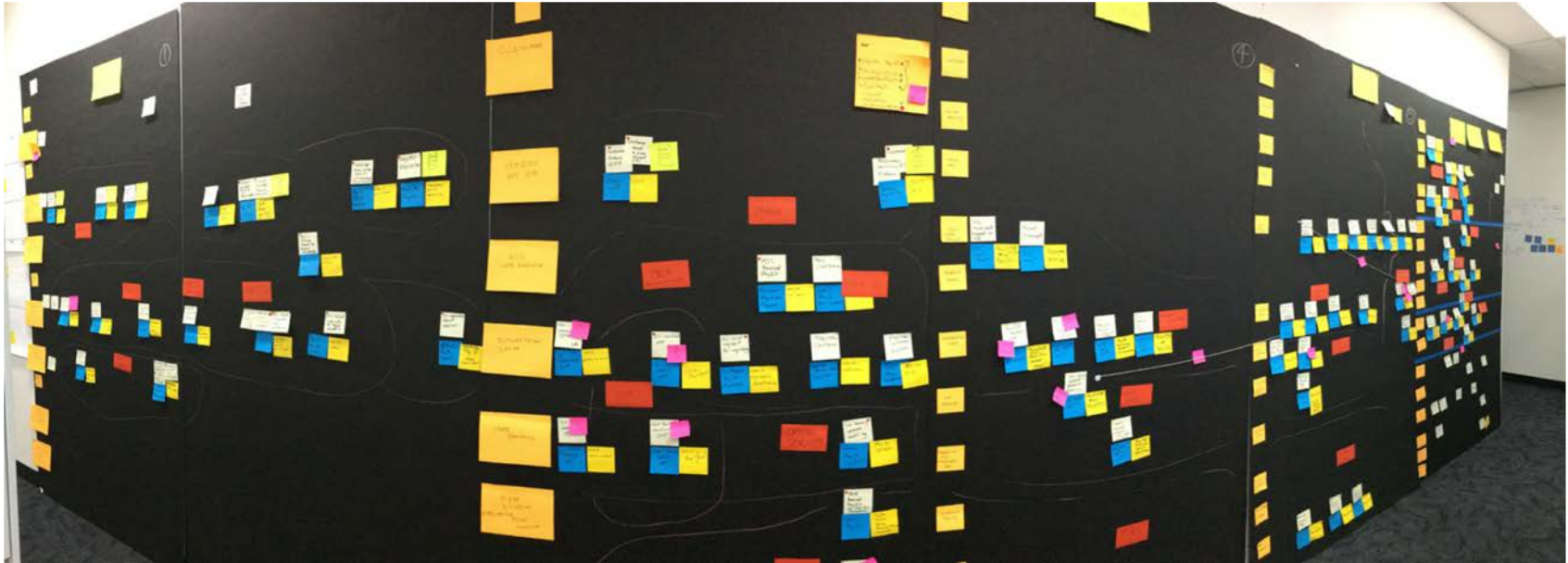
Using the NPP use case I will cover off how to do event storming in the steps we use for a particular process flow (Registering a PayID):

1. Specify the scenario to be mapped
2. Identify all actors, including human and non-human
3. Map events that happened
4. Include commands
5. Identify data required
6. Identify views required
7. Determine manual processes
8. Note unknowns
9. Identify unhappy paths
10. Group events into bounded contexts
11. Add NFRs (Non-Functional Requirements)
12. Create an MVP



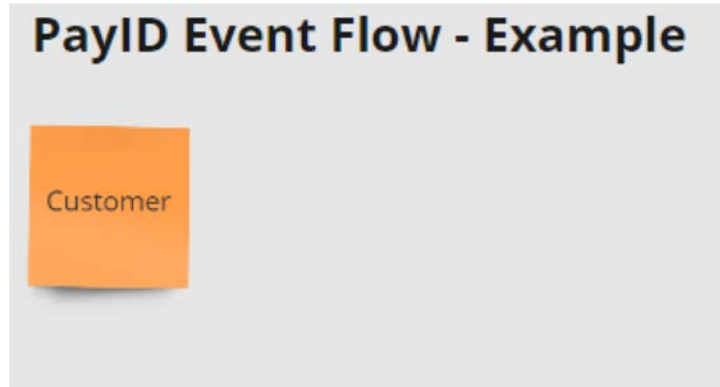
# Event Storming – NPP Case Study

- Scenario:
  - Register PayID
  - Happy Path
  - Existing Customer



# NPP Case Study – Identify Actors

- Actors:
  - Any human or system involved in the process
  - Starting with PayID Event Flow, have Customer



# NPP Case Study – Identify Actors

- Go through and include all actors

## PayID Event Flow - Example

Customer



## PayID Event Flow - Example

Customer

App

# NPP Case Study – Identify Actors

## PayID Event Flow - Example

Customer

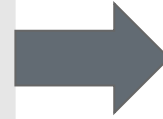
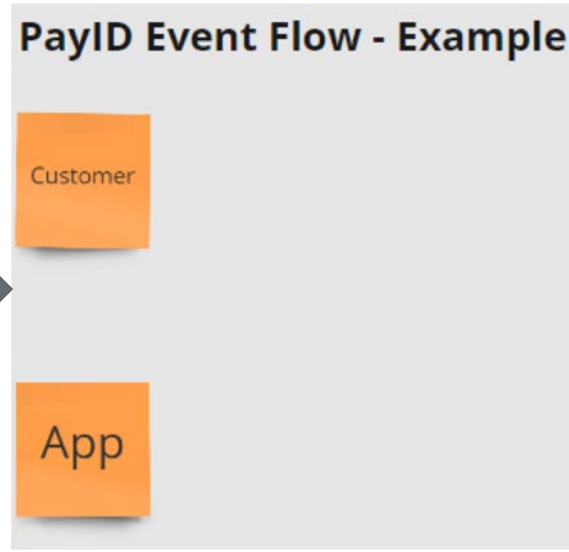
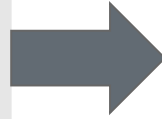
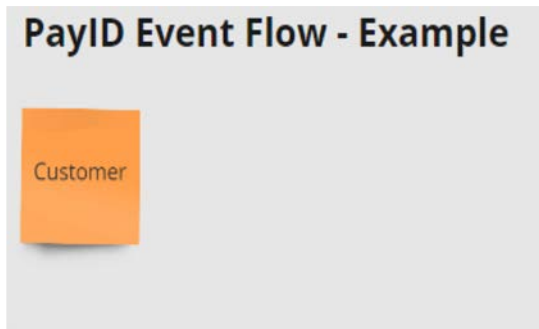
App

Core  
Banking

ACL

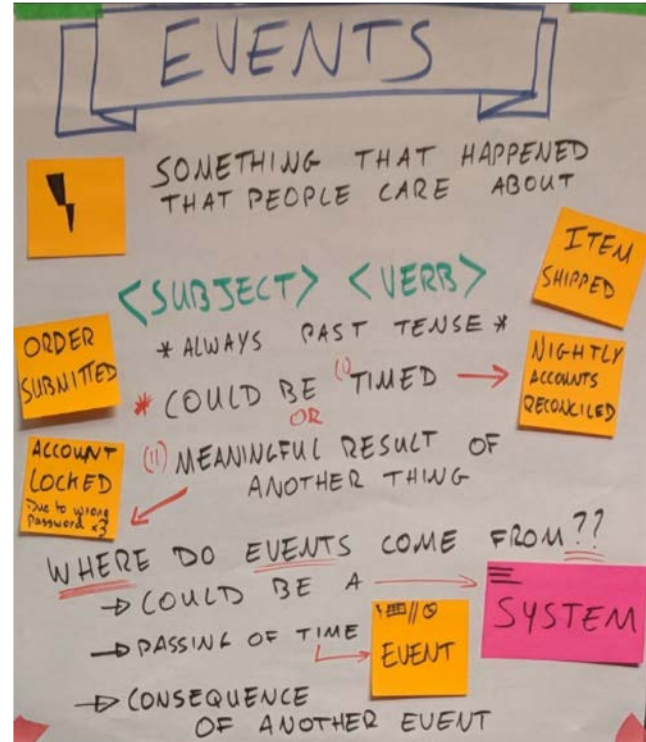
Fraud  
System

- Go through and include all actors



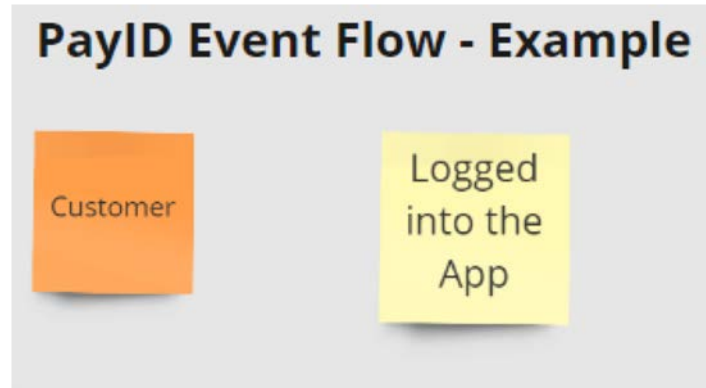
# NPP Case Study – Define Events

- Events:
  - An action that occurred in the system (or process)
  - May occur at a specific time
  - Write each domain event with a few words and a verb
  - Written in past tense on a sticky note
  - Describe what happened
  - Events must be worded in a way that:
    - Is meaningful to domain experts and business
    - Explains what happens in business terms
      - Not what happens inside the system
  - Place all the events in sequence



# NPP Case Study – Write the First Event

- Write the first Event:
  - Customer Logged into the App and
- Place it against the relevant actor
  - In this instance, against Customer



# NPP Case Study – Define Events

- Events:
  - Continue writing out all events

## PayID Event Flow - Example

Customer

Logged  
into the  
App

App

Selected  
Payment  
Menu

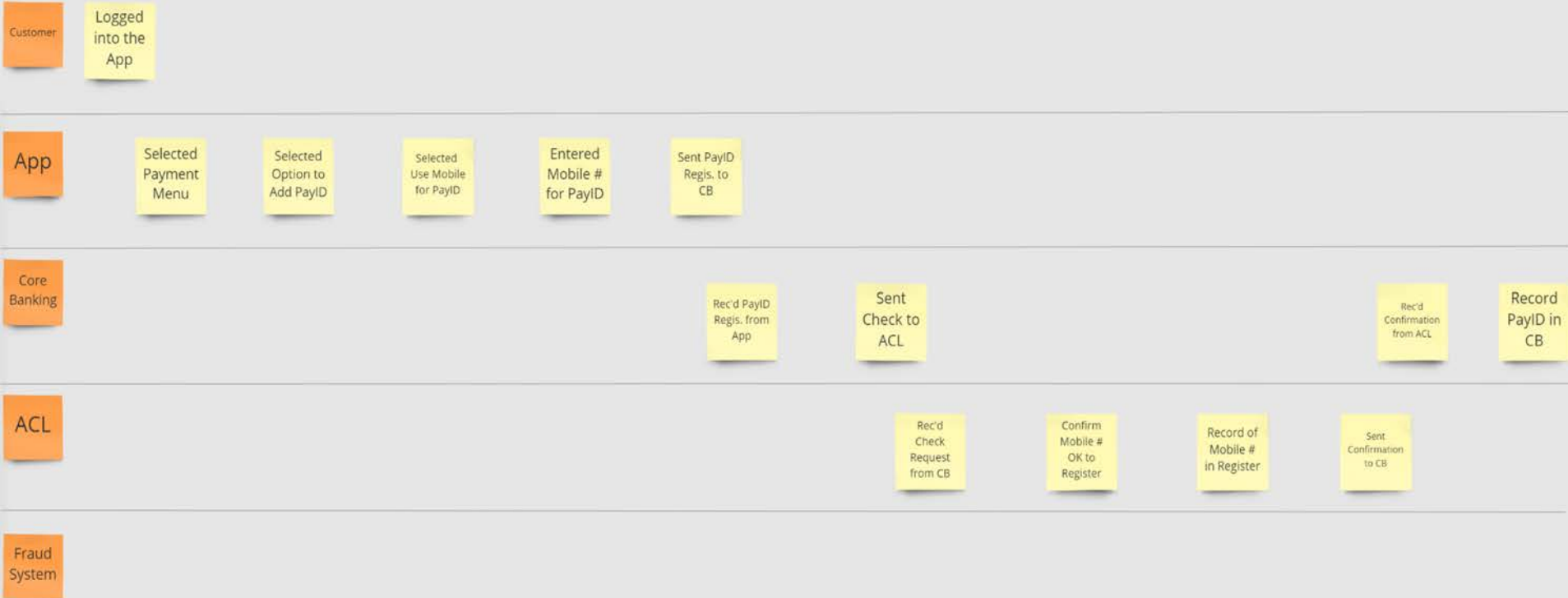
Selected  
Option to  
Add PayID

Selected  
Use Mobile  
for PayID



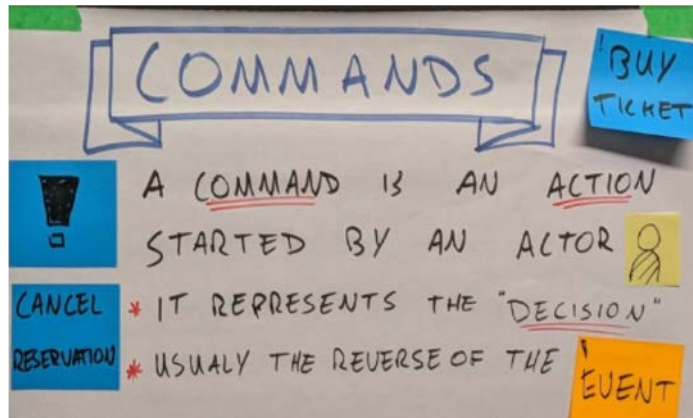
# NPP Case Study – Define Events

## PayID Event Flow - Example



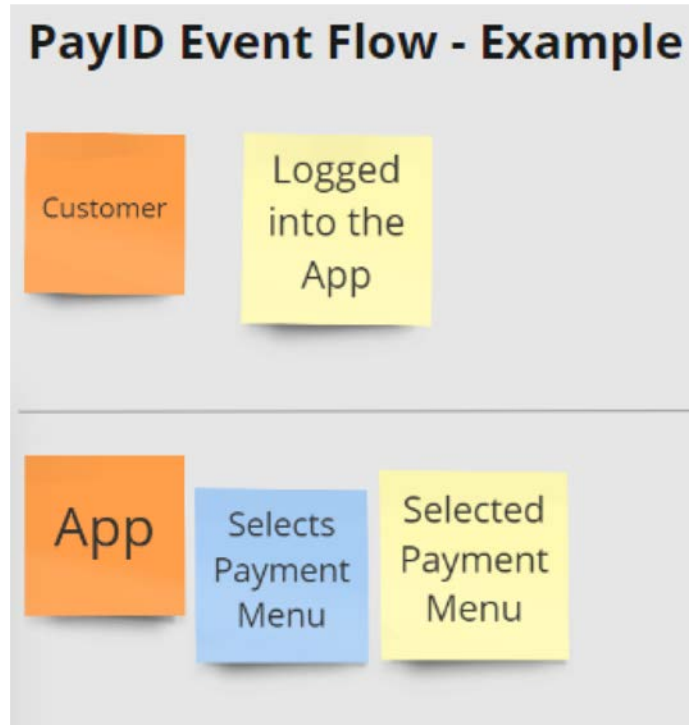
# NPP Case Study – Add Commands

- Commands are the result of a user decision or actor action
- Represents an action or intent
- Present tense
- Things that trigger an event
- Usually the reverse of the event
- To understand a command, need to know:
  - Who starts a command (actors)
  - Information needed to allow the command to run
- Provide a natural context for microservices



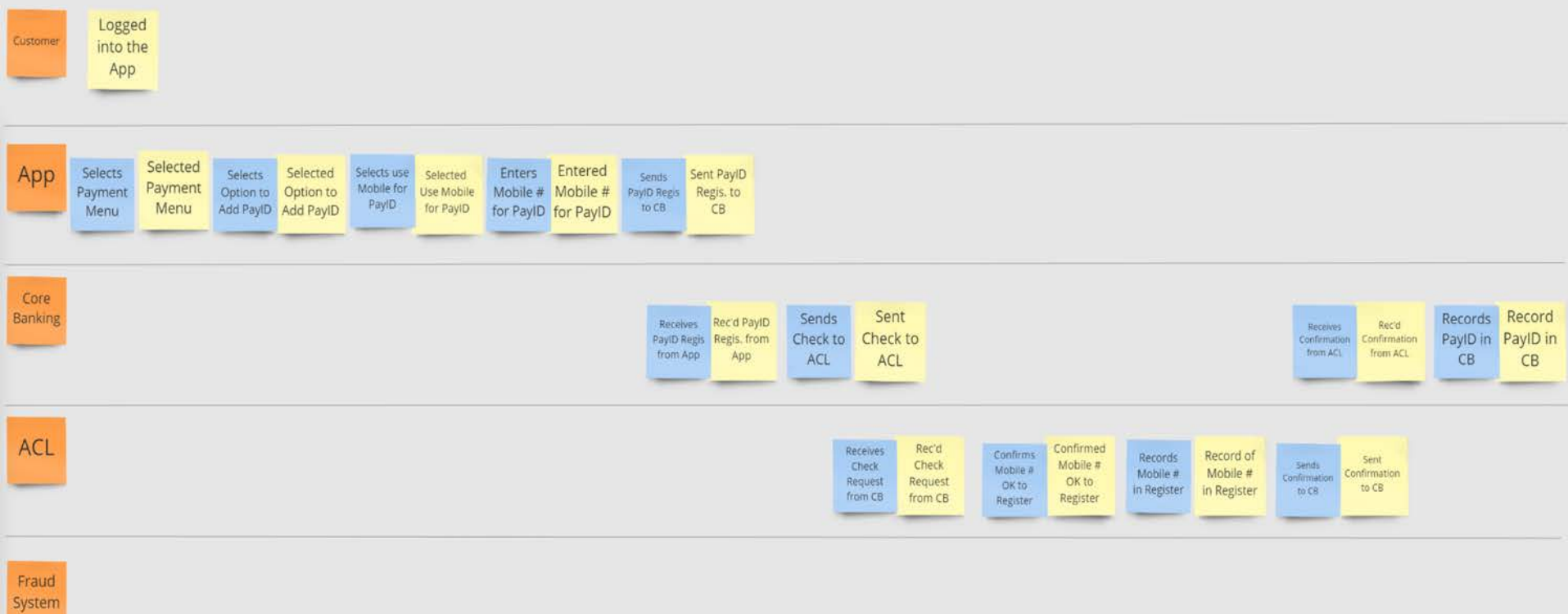
# NPP Case Study – Add Commands

- Include Commands for each Event - write in present tense



# NPP Case Study – Add Commands

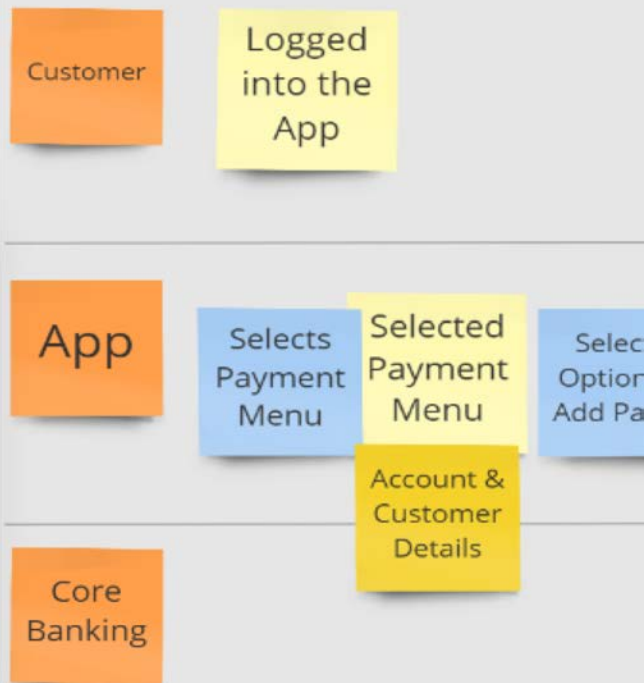
## PayID Event Flow - Example



# NPP Case Study – Add Data

- Actors consume data through a user interface and interact with a system by issuing commands
- Data points are added to events
  - Identifies information required for each event
- Generally done at a high-level
  - Account Details, Customer Name, etc
- Data required is specific to the individual event
- Represents how data flows between systems

## PayID Event Flow - Example



# NPP Case Study – Add Data

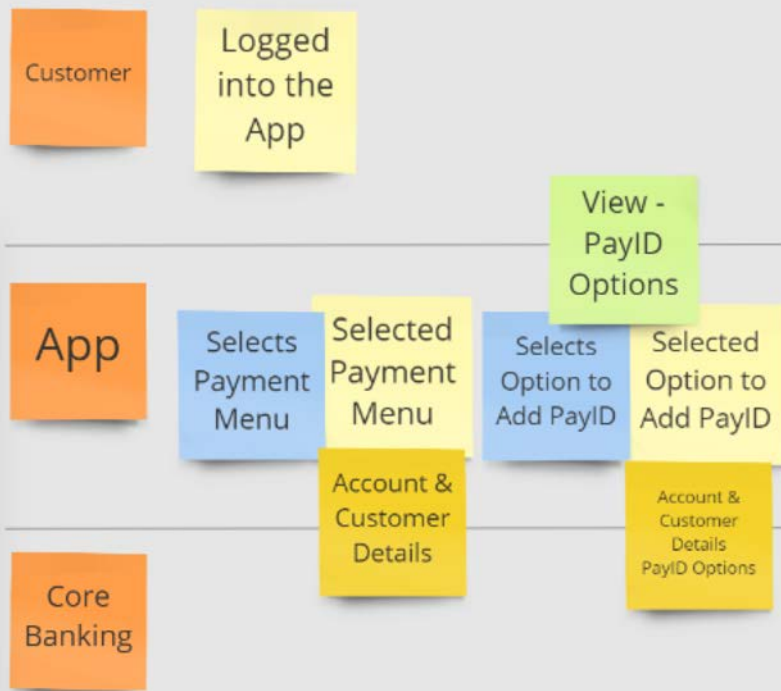
## PayID Event Flow - Example



# NPP Case Study – Add Views

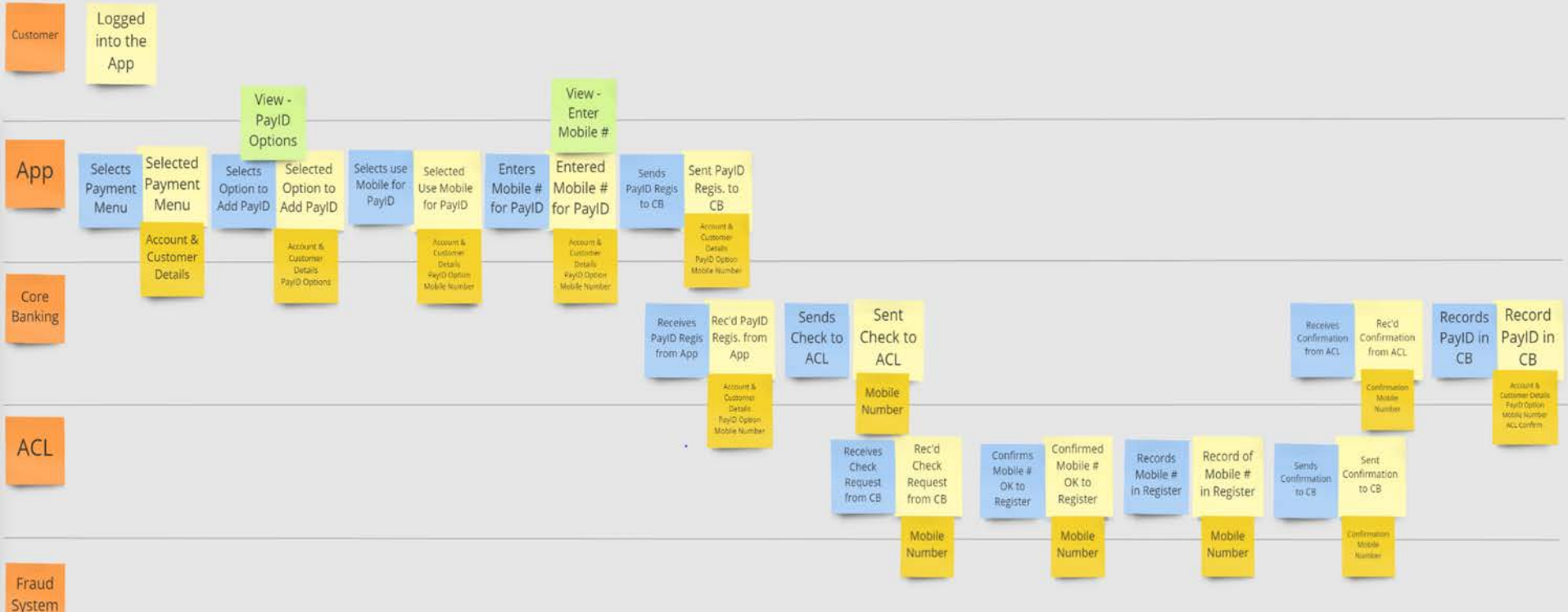
- Views:
  - Indicates the need to create a UI
  - Examples:
    - Mobile App screen
    - New screen in a system
- Anytime an actor needs to see / view something it is flagged as a view to be created
- Name the view something that makes sense

## PayID Event Flow - Example



# NPP Case Study – Add Views

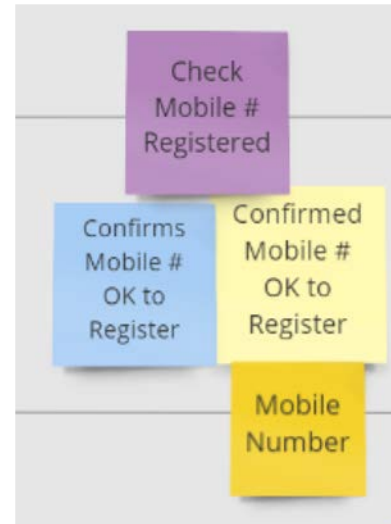
## PayID Event Flow - Example





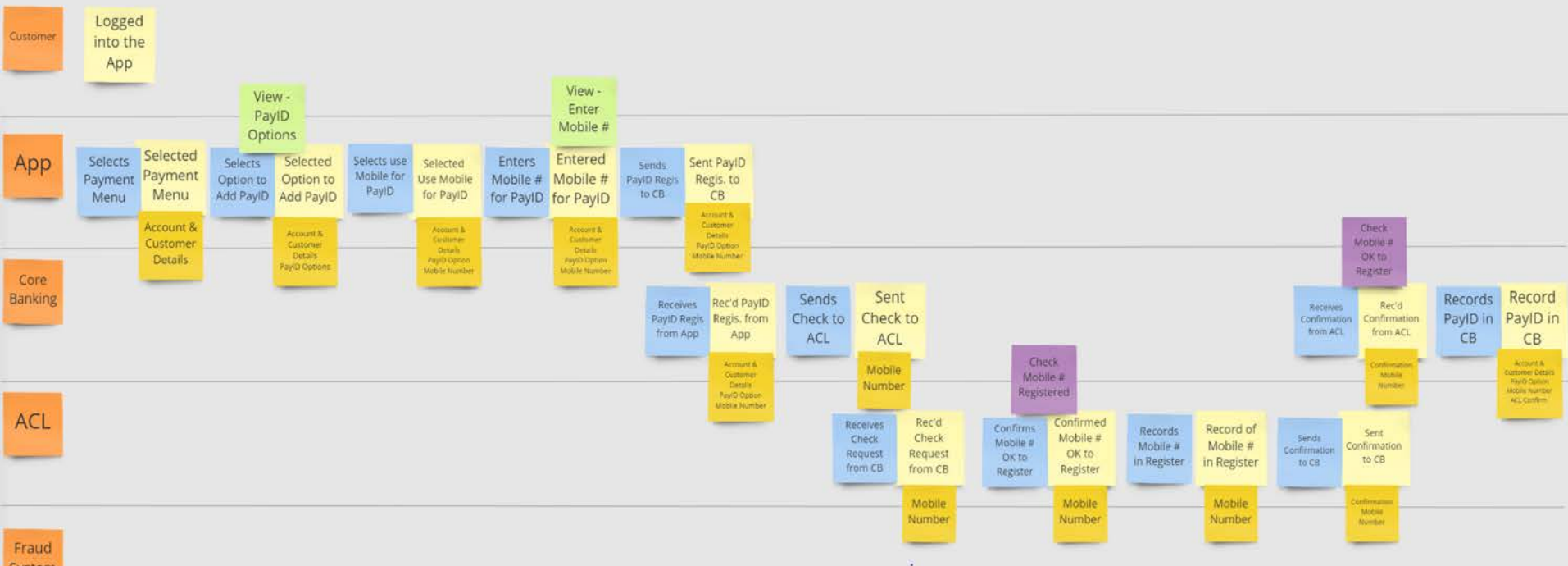
# NPP Case Study – Add Processes

- Processes are reactive logic
- They take place after an event occurs
- Generally indicated by light purple sticky notes
- Processes can be either:
  - A manual step a human follows
  - It may be an automated step
- Manual steps may include a:
  - Documented procedure that needs to occur
  - Decision that needs to be made



# NPP Case Study – Add Processes

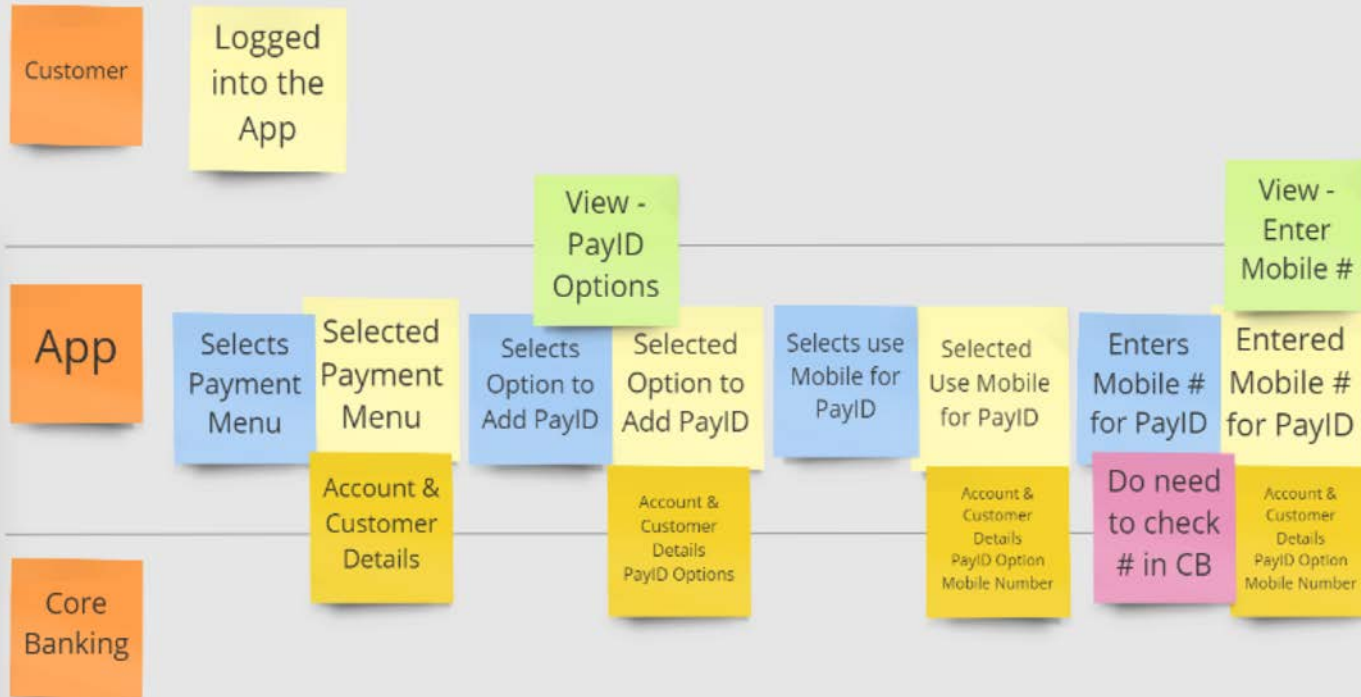
## PayID Event Flow - Example



# NPP Case Study – Identify Unknowns

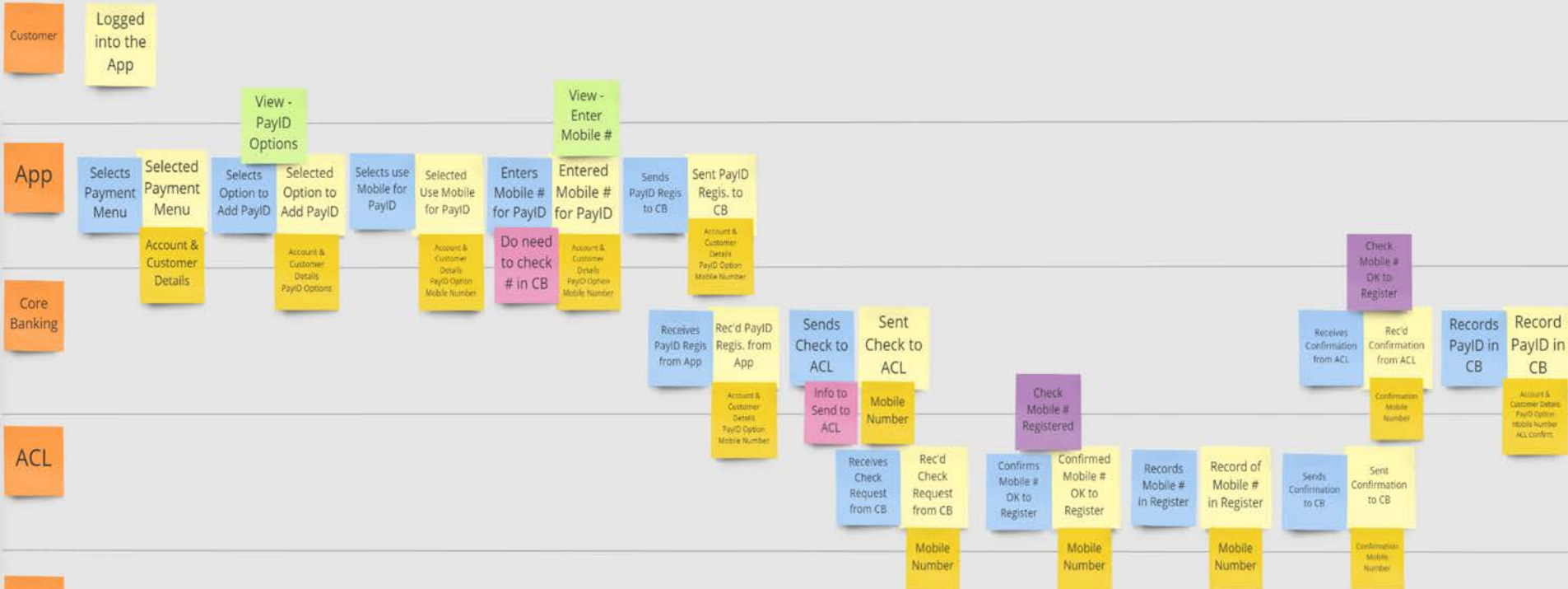
- Make a record of anything requiring clarification
- Referred to as Unknowns
- Use a pink sticky note to identify
- Become a Business or Tech Spike which is a user story to investigate in Sprint 0 or early sprints
- Useful to keep conversation moving

## PayID Event Flow - Example



# NPP Case Study – Identify Unknowns

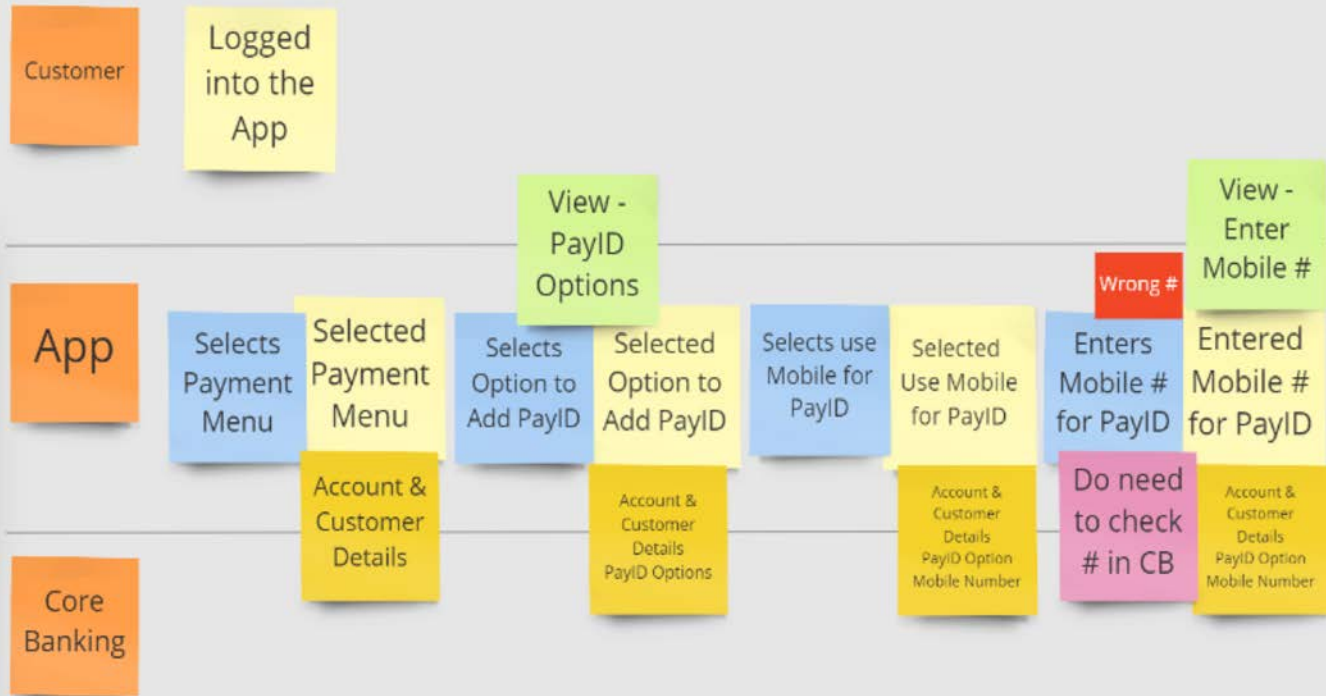
## PayID Event Flow - Example



# NPP Case Study – Identify Unhappy Paths

- Using small coloured sticky, identify the Unhappy Paths
- A record should be kept of these
- Future event storming sessions may need to be run to understand Unhappy Path requirements

## PayID Event Flow - Example



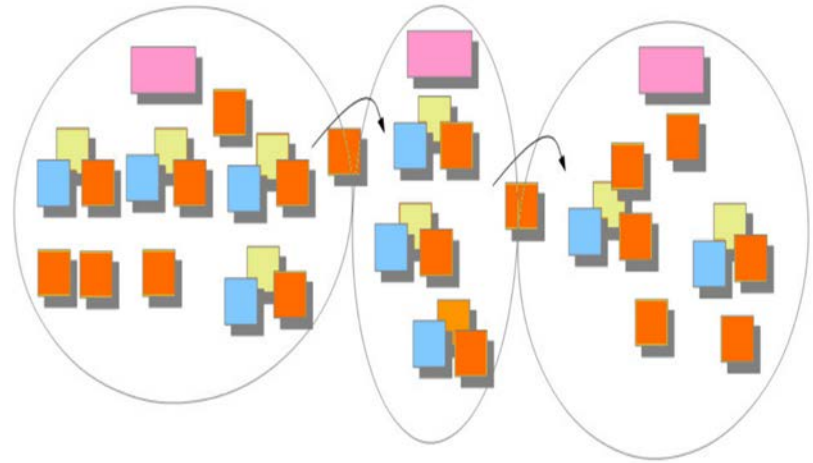
# NPP Case Study – Identify Unhappy Paths

## PayID Event Flow - Example



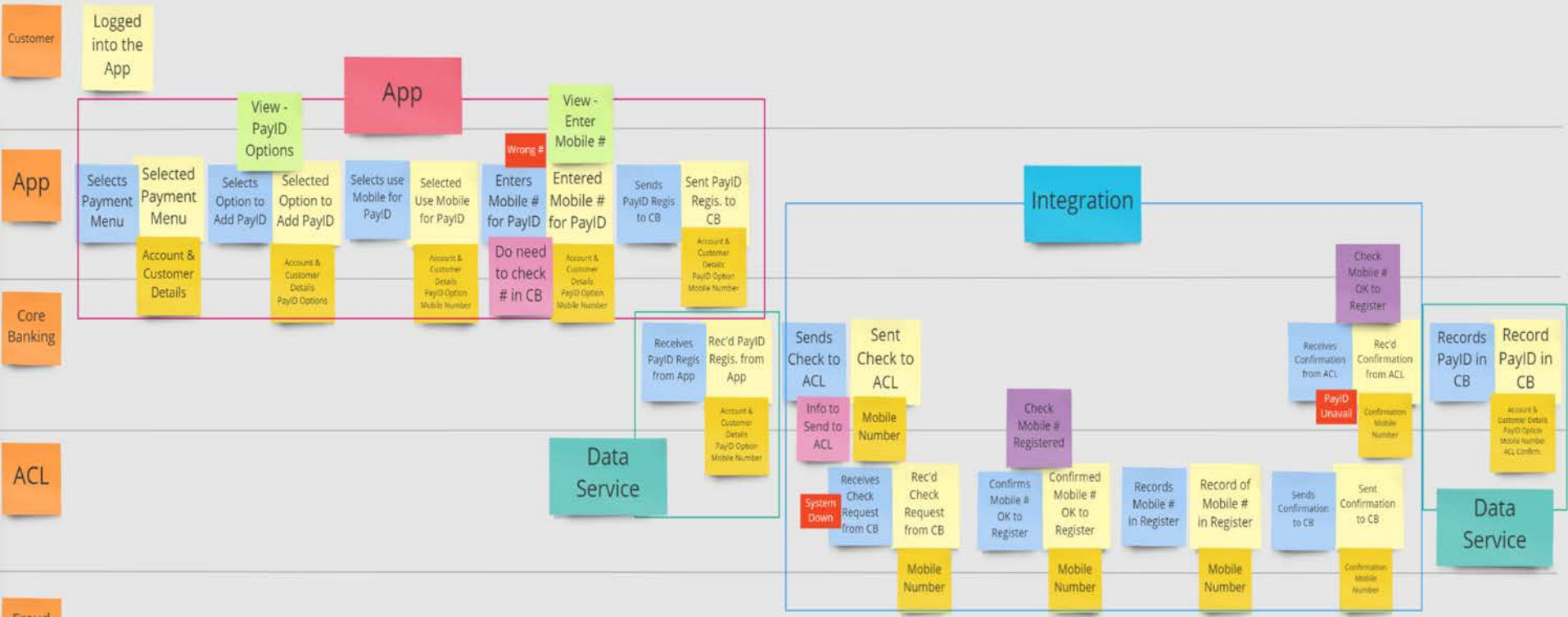
# NPP Case Study – Bounded Context

- Bounded Context & How We Use It:
  - Align to The Open Group's Technical Reference Model (TRM)
    - Group bounded contexts by:
      - App (Digital Access Channel)
      - Data
      - Integration
      - Enterprise Applications
  - Use to identify which teams are required to work on specific aspects of the flow / solution



# NPP Case Study – Bounded Context

## PayID Event Flow - Example

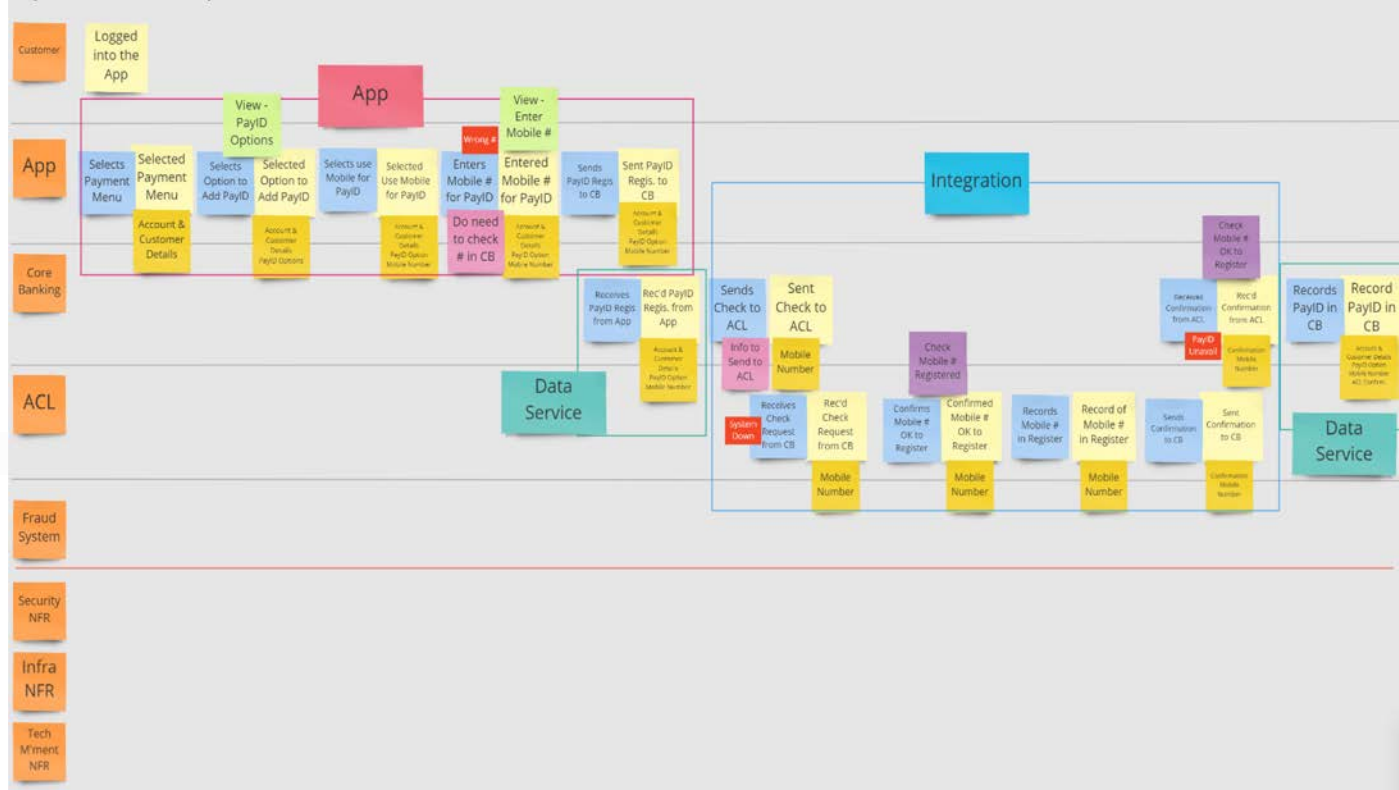




# NPP Case Study – Extended to Include NFRs

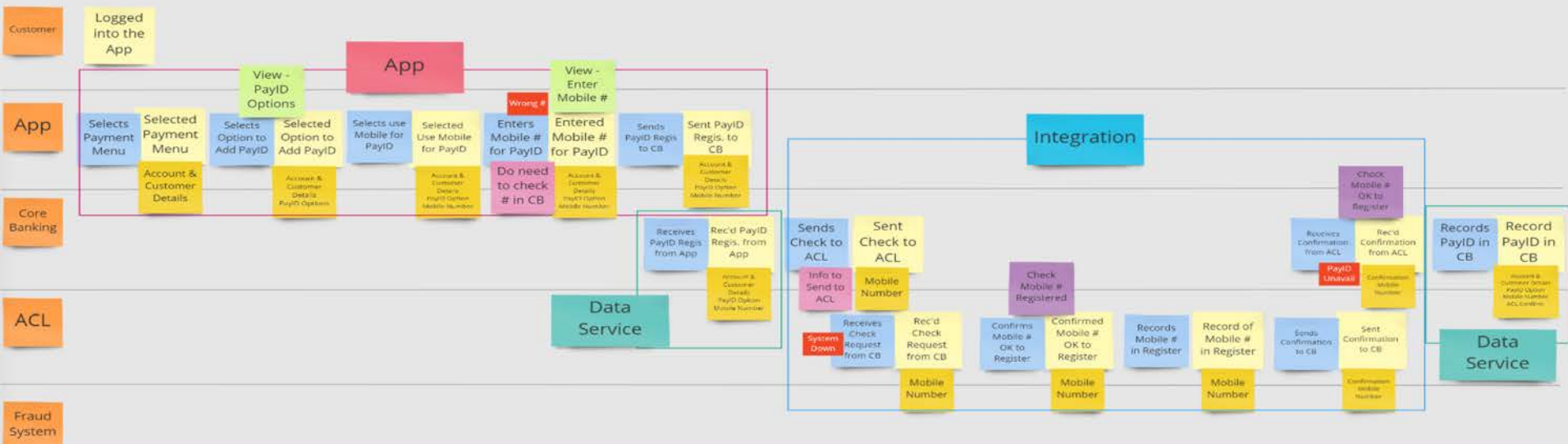
- Extended Event Storming to capture NFRs:
  - Use the following NFR groups:
    - Security
    - Integration
    - Technology Management
  - Tech team identifies the relevant NFRs for each NFR “actor” for each event
  - Provides more holistic picture of requirements

PayID Event Flow - Example



# NPP Case Study – NFRs

## PayID Event Flow - Example



Security NFR	S NFR 1	S NFR 2	S NFR 3	S NFR 4	S NFR 5
Infra NFR	I NFR 1	I NFR 2	I NFR 3	I NFR 4	
Tech M'ent NFR	TM NFR 1	TM NFR2	TM NFR 3	TM NFR4	TM NFR 5

# NPP Case Study – Define MVP

- Go through and number each event in the event flow, starting at 1
- Rewrite each event (include the sequence number)
- Write out each Bounded Context
- Place these as headings on a separate board
- Place all events under the relevant Bounded Context heading

PayID Event Flow - Example



# NPP Case Study – MVP

## PayID MVP - Example

App

Data Service

Integration

MVP

Selected Payment Menu

Selected Option to Add PayID

Selected Use Mobile for PayID

Entered Mobile # for PayID

Record PayID in CB

Rec'd PayID Regis. from App

Sent Check to ACL

Rec'd Check Request from CB

Confirmed Mobile # OK to Register

Confirmed Mobile # OK to Register

Record of Mobile # in Register

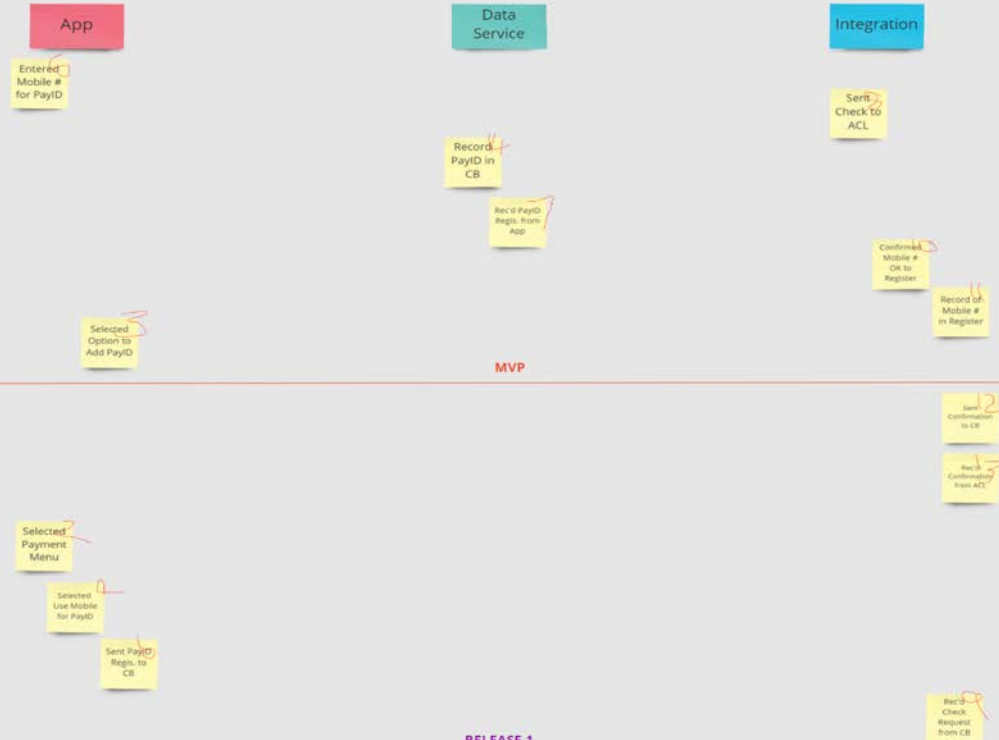
Sent Confirmation to CB

Rec'd Confirmation from ACL

# NPP Case Study – Define MVP

- Prioritise each event, starting with the most important
  - Take a vertical or horizontal slice
  - Depends on what is most important
- Move the sticky-notes up
- Keep only the most important for the MVP
- Can include stretch goals or subsequent releases
- Priorities are mutually exclusive
- Numbers on events should align with those for the same event in the event flow

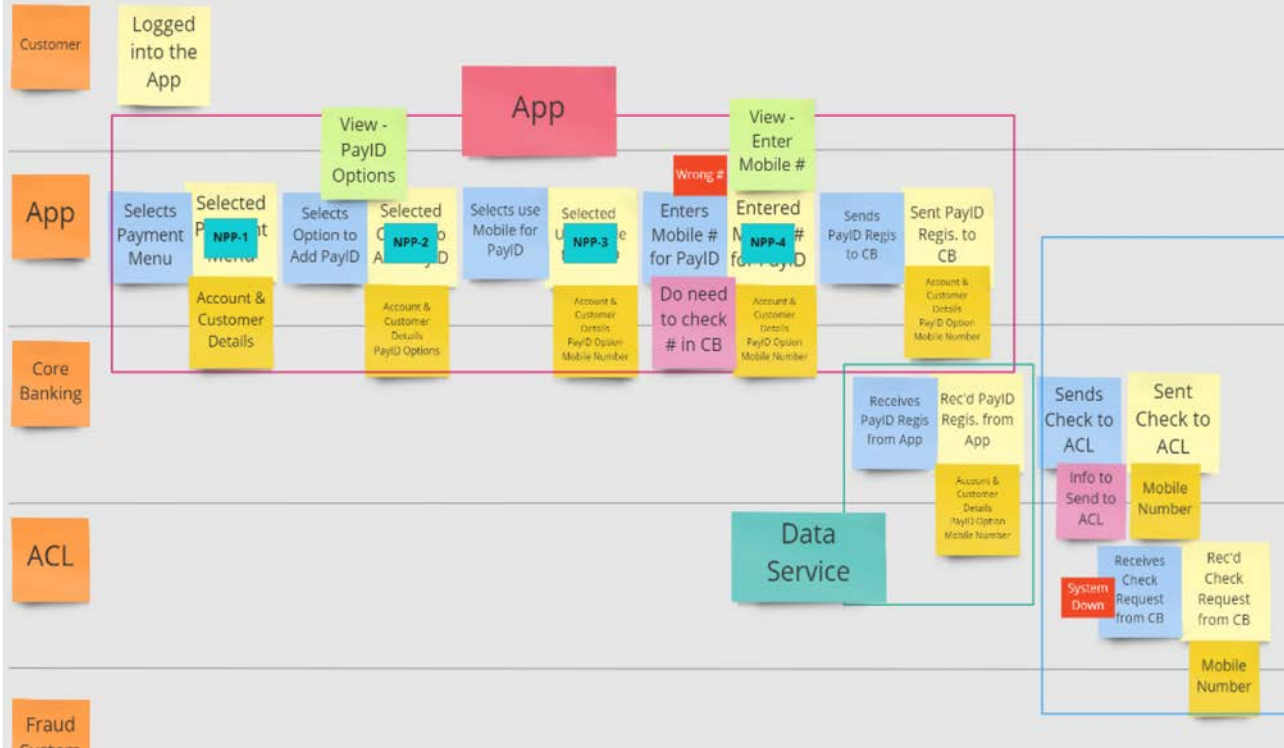
PayID MVP - Example



# NPP Case Study – User Stories

- Each event becomes a user story
- May need to split or combine them
- Add tech sub-tasks & include NFRs
- Allocate to the relevant team
- Keep a record in Confluence
- Place small sticky-notes with Jira ID# onto each event so don't duplicate them
- Create a digital copy of the event flow

PayID Event Flow - Example



# Who is Needed to Run a Successful Session

- Invite the Right People!
- Works best when you have the entire team involved
  - Technical and business
  - Start the process together
- Tried it with just business to identify events then brought in the technical team
  - Takes longer than creating it together
- Decision-maker or delegated authority
- Typically include:
  - Developers
  - Architects
  - UI/UX
  - Domain experts (SMEs)
  - Product Owner
  - Scrum Master
- Anyone who has a role in creating the solution or with required knowledge should be included



# Questions or Challenges in Facilitation

## Questions:

- Can I do it electronically?
  - You can try - its painful!
- Do I need a lot of wall space?
  - Yes - use a meeting room / movable boards

## Challenges in Facilitating:

- Focus on what not how
  - No solutioning
- Focus on Happy Path initially
  - Otherwise you won't finish!
- Useful to have a decision-maker or delegate





# Timeframes

- Timeframes to run a session:
  - Small process of  $\leq 30$  events
    - $\frac{1}{2}$  day
  - Medium process of  $> 30 \leq 60$  events
    - 1 day
  - Large process of  $> 60$  events
    - 1-2 days
- More often you and team do it, faster you become
  - Overtime you'll find efficiencies



# Lessons Learned

- Define key terminology upfront
- Ensure have defined scope of work & limiting assumptions:
  - Example:
    - Scenario: Register PayID
    - Assumptions: Happy Path & Existing customer
- Focus on happy path but identify unhappy paths
- Use a parking lot
- Take photos & document in Confluence
- Put small stickies over events once written user story
- Keep event flow visual on the wall if possible
- Saves time & reduces need for detailed requirement documents
- Standing & participation session
- Get team to write-out the sticky notes



# Take-Away – Start Today!

- Start Today:
  - Pick a process that you think may need to be changed or fixed
  - Identify all the actors involved
  - Start by mapping out the processes
  - Use it as a starting point and re-iterate
  - Share it with someone else and get their feedback
  - Try it with a small group on a small project

# Useful Resources

- The following resources will be very helpful:
  - <https://www.eventstorming.com/>
  - <http://ziobrando.blogspot.com/2013/11/introducing-event-storming.html>
  - <https://servicesblog.redhat.com/2017/04/17/accelerate-application-development-with-event-storming-and-open-innovation-labs/>
  - <https://openpracticelibrary.com/practice/event-storming/>
  - <https://www.linkedin.com/pulse/using-event-storming-practice-heritage-bank-sandra-arps/>
  - <https://www.linkedin.com/feed/update/urn:li:activity:6527384658114674688>
  - [https://en.wikipedia.org/wiki/Event\\_storming](https://en.wikipedia.org/wiki/Event_storming)
  - <https://techbeacon.com/devops/introduction-event-storming-easy-way-achieve-domain-driven-design>
  - <https://medium.com/@springdo/a-facilitators-recipe-for-event-storming-941dcb38db0d>
  - <https://medium.com/@springdo/a-facilitators-recipe-for-event-storming-941dcb38db0d>