

# Table of Contents

Long-Running User Flows (10 pages) . . . . .	1
Continuing our design journey . . . . .	1
Implementing Sagas . . . . .	1
Taking Care of Deadlines . . . . .	2
Distributed Exception Handling . . . . .	2
Keeping Track of the Overall Flow . . . . .	2
Deciding between Orchestration and Choreography . . . . .	2

## Long-Running User Flows (10 pages)

In the long run, the pessimist may be proven right, but the optimist has a better time on the trip.

— Daniel Reardon

Not all capabilities can be implemented in the form of a simple request-response interaction, requiring the management of complex state and interactions either with external systems or human-centric operations or both. In other cases, there may be a need to perform business logic at a certain time in the future. In this chapter, we will look at implementing both long-running user operations (sagas) and deadlines. We will also look at how we can keep track of the overall flow using log aggregation and distributed tracing. We will round off by looking at when/whether to choose explicit orchestrations components or implicit choreography.

## Continuing our design journey

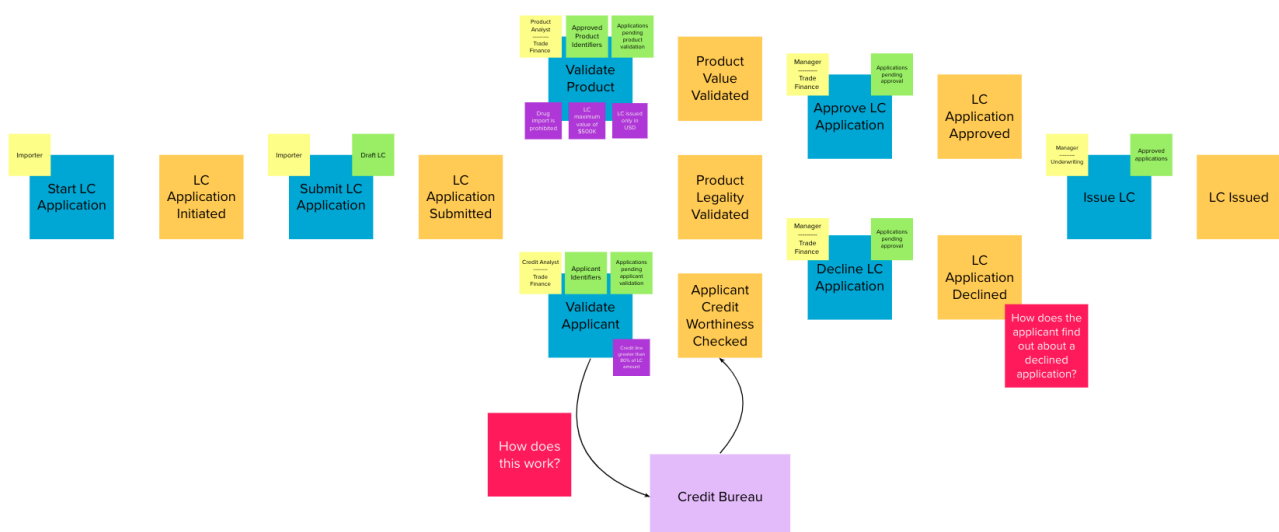


Figure 1. Recap of eventstorming session

## Implementing Sagas

**Taking Care of Deadlines**

**Distributed Exception Handling**

**Keeping Track of the Overall Flow**

**Deciding between Orchestration and Choreography**