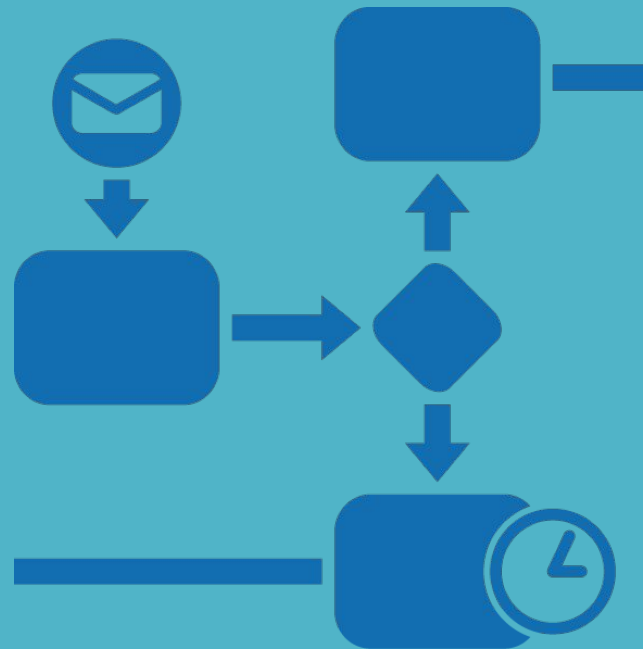


Donothrow

PROCESS AND SERVICE DESIGN PROJECT



Our service



Our customers

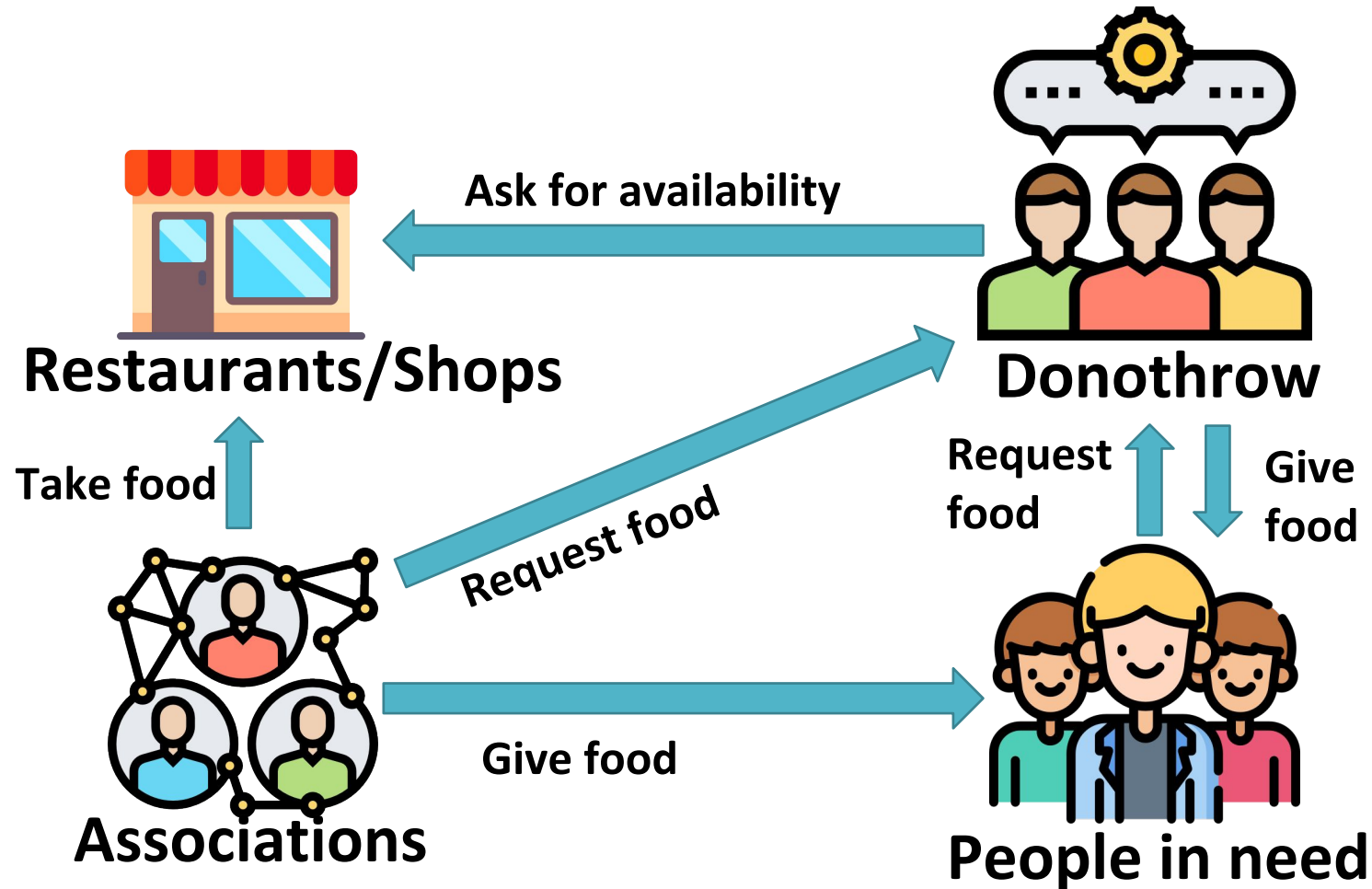
Associations helping people in need



People in need directly



The big picture



Steps

Blueprints

BPMN (High-Level, Choreography, Orchestration)

Process Verification (Petri Nets)

REST API Portfolio Design (Swagger)

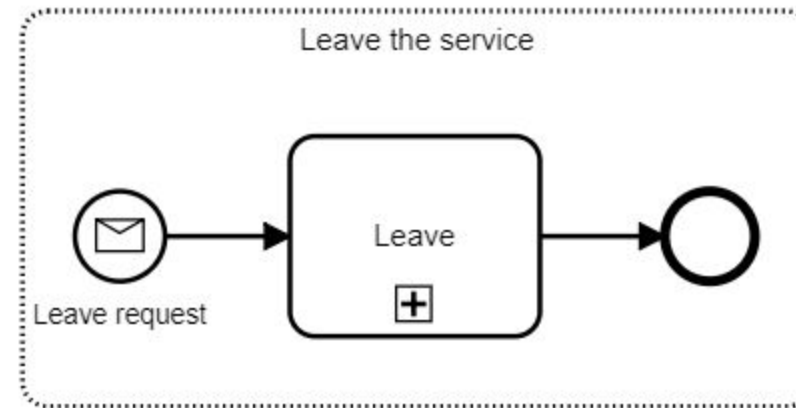
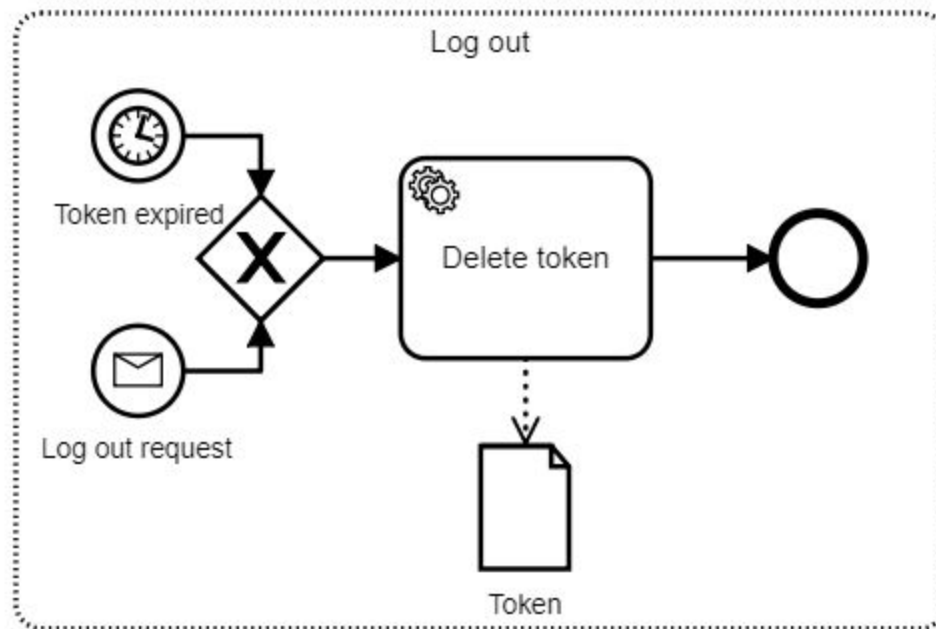
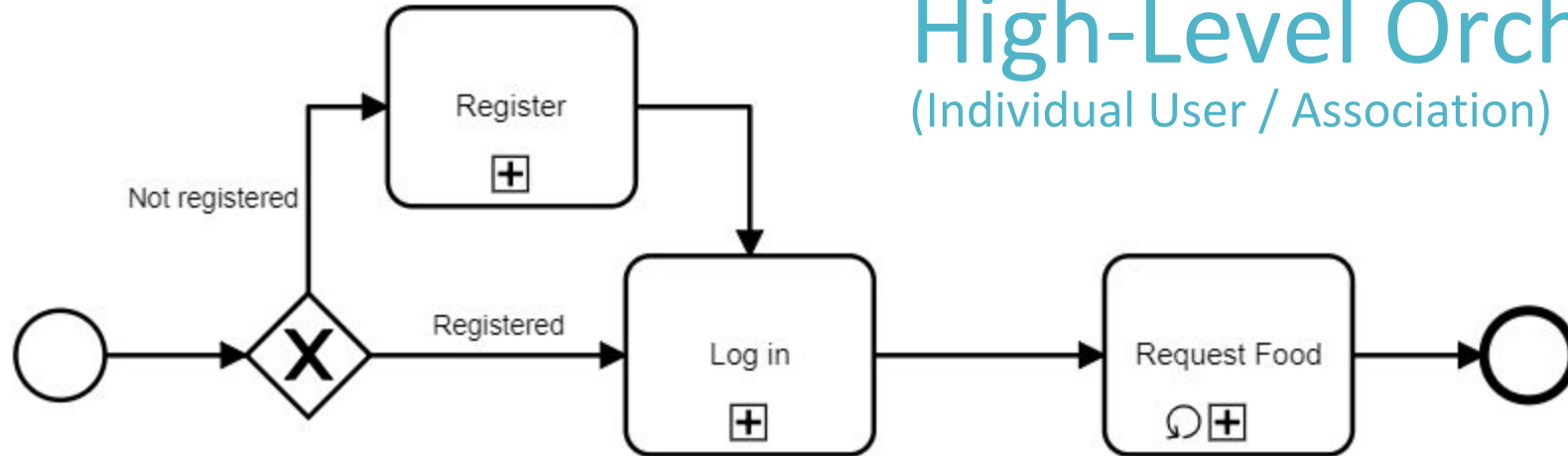
REST API Implementation/Deployment (Node + Heroku)

Process Execution (Camunda + Java)

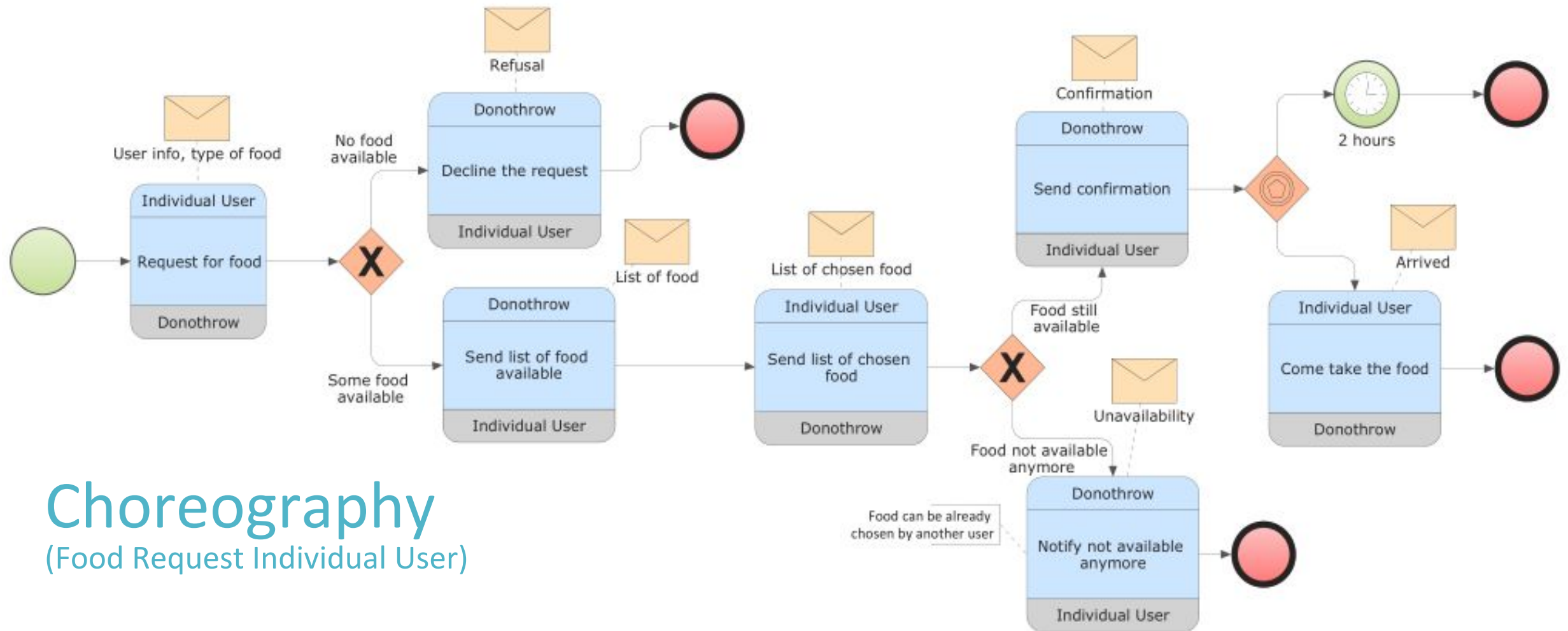
		AWARE			JOIN			USE			LEAVE		
User	Step	Surfs on the Internet	Receives leaflets	Visits our social networks	Downloads the app	Registers on app or website	Logs in on app or website	Requests for food	Receives a notification	Confirms the request	Fills form about reasons of the departure	Deletes account	Receives email of confirmation
Channels	Face to face		Presentation of Donothrow										
	Print		Link to the website, QR Code for the app										
	Web	Homepage of the website with information on Donothrow		Presentation of Donothrow, link to the website		Fills a form with personal characteristics (professional status...)	Inserts email and password	Inserts user info and the type of food needed	Reads the notification (if the request was accepted, what food is available)	Chooses some food from the proposed list	Checks some reasons of departure in a list	Clicks a button	
	Email												Reads the email
	Mobile			Presentation of Donothrow, link to the website		Fills a form with personal characteristics (professional status...)	Inserts email and password	Inserts user info and the type of food needed	Reads the notification (if the request was accepted, what food is available)	Chooses some food from the proposed list	Checks some reasons of departure in a list	Clicks a button	
Backstage processes	IT-Department					Database: insertion of a new user	Database: verification that user exists	Database: creation of a new request		Database: update of the request		Database: deletion of the user	
	Third party		A printing company		Google Play Store				Amazon Simple Notification Service				Gmail

High-Level Orchestration

(Individual User / Association)



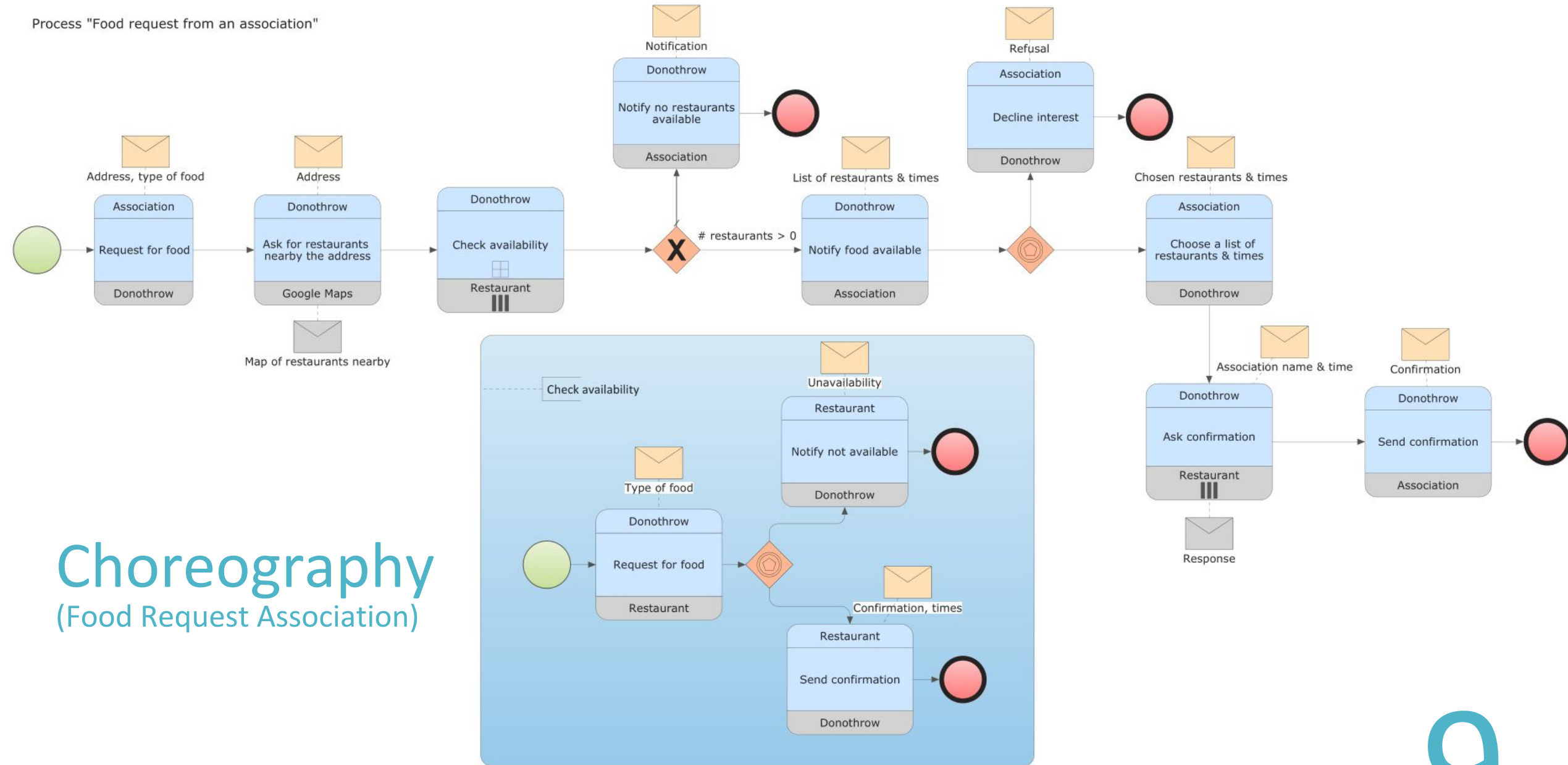
Process "Food request from an individual user"



Choreography

(Food Request Individual User)

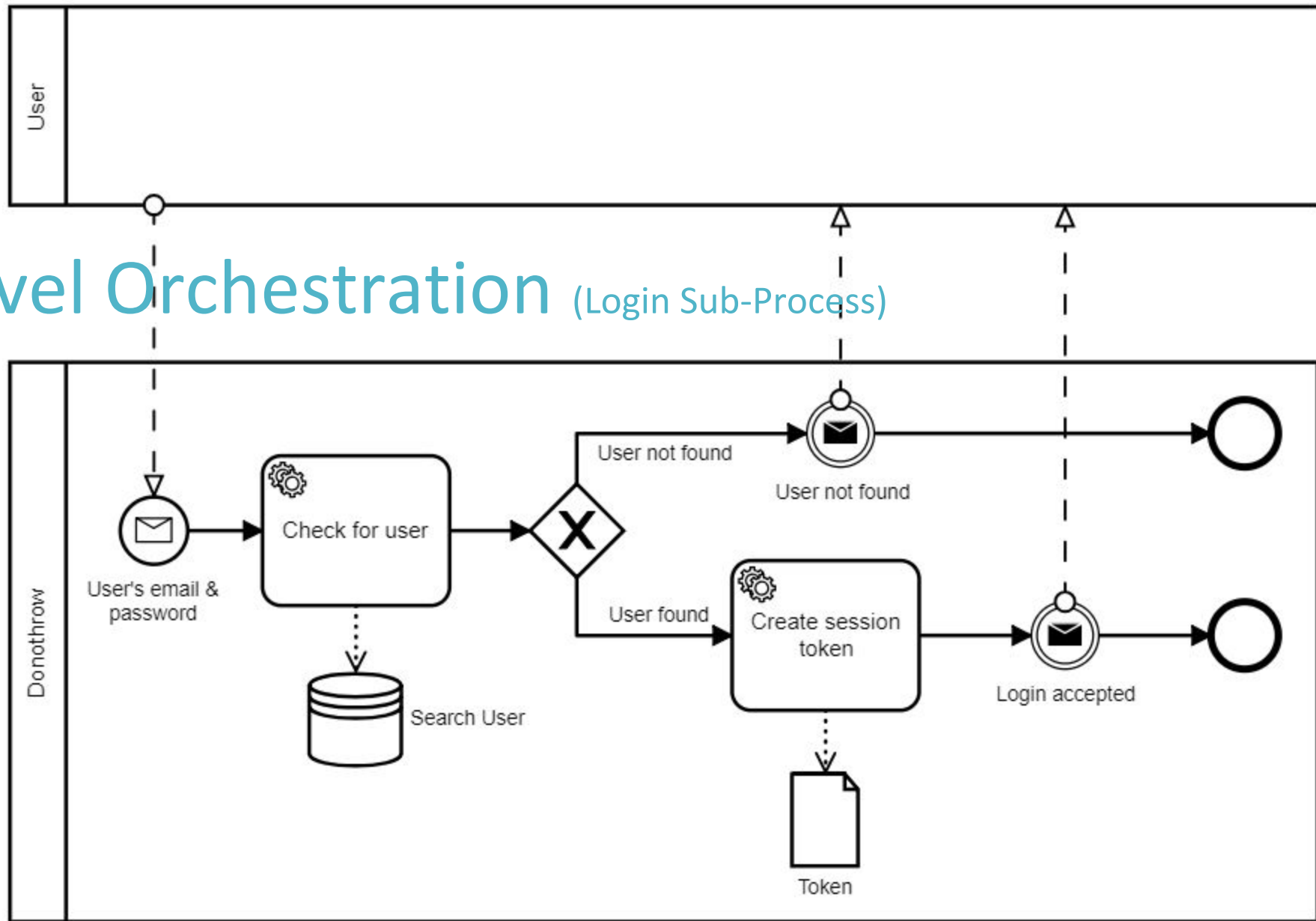
Process "Food request from an association"



Choreography

(Food Request Association)

Low-Level Orchestration (Login Sub-Process)



Much more was modelled!

CHOREOGRAPHY

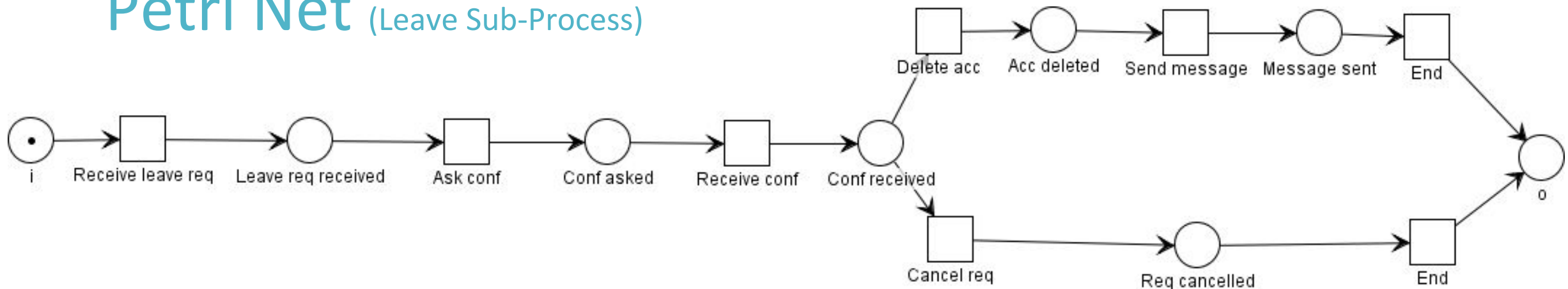
- Individual User Food Request
- Association Food Request
- [Donothrow Food Request]
- Register
- Login
- Leave

ORCHESTRATION

- High-Level Individual User / Association
- [High-Level Donothrow]
- Low-Level Register
- Low-Level Login
- Low-Level Leave (soundness verification)
- Low-Level Individual User Food request (executed)

Process Verification

Petri Net (Leave Sub-Process)



Process is sound iff :

- The corresponding Petri Net is a Workflow Net ✓
- For any case the process terminates in o with only tokens in o ✓

REST API Portfolio Design

food Everything about food	
POST	/food/ Add a new food to database
GET	/food/type/{type} Finds food by type
PUT	/food/{id} Update an existing food items by id
GET	/food/{id} Finds food by id
DELETE	/food/{id} delete food by id

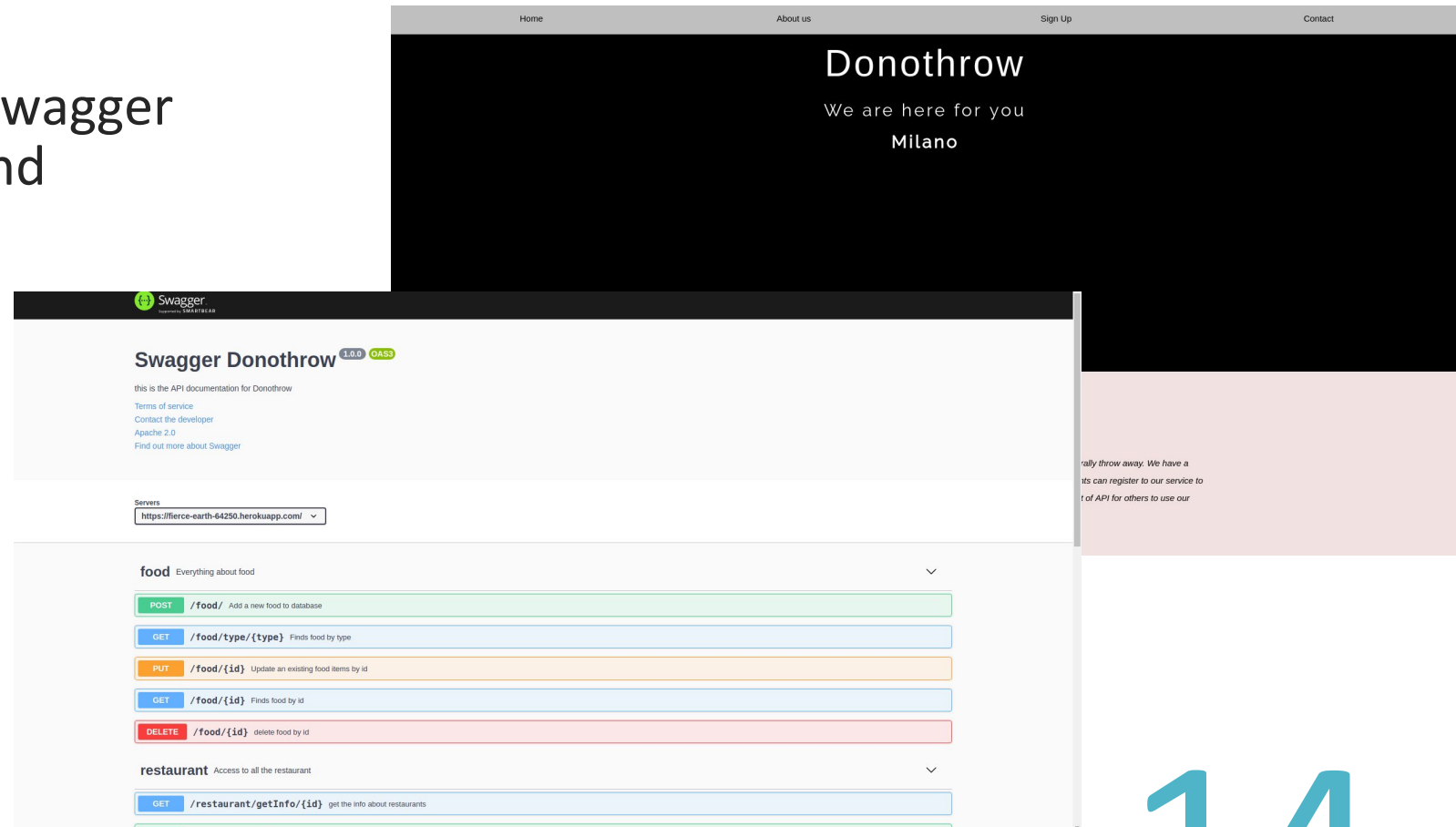
request Access to all the request	
GET	/request/{id} find request by id
DELETE	/request/{id} delete request by id
GET	/request/all find all the request
POST	/request add new request

user Operations about user	
GET	/user/info/{id} get user info by id
PUT	/user/info/{id} update user information
POST	/user/signUp register user
POST	/user/signIn signIn with user info

restaurant Access to all the restaurant	
GET	/restaurant/getInfo/{id} get the info about restaurants
POST	/restaurant/add add new restaurants

REST API Implementation/Deployment

- Nodejs server
- API documentation with swagger
- HTML and CSS for front end
- Deployed in heroku



Execution

Time for a demo!



Challenges

Synchronization of the **work**

Solution: Frequent Reviews in Group

Design a **reusable API**

Solution: Apply the REST properties

Execute a process with **unknown technologies** (Camunda/Java)

Solution: Watch tutorials from Camunda