Map

2026-11-19 · `examples/types/map.json`

2 views · 17 cards · 18 lines

Contents

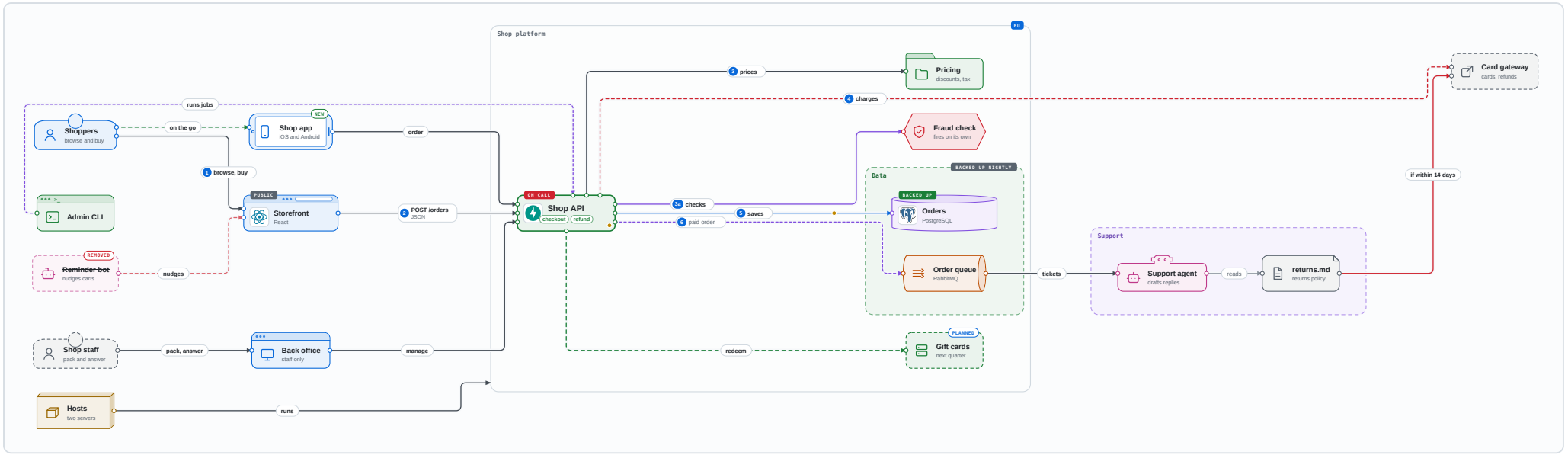
The bookshop's systems 2
Who uses the shop, what runs it, and where an order goes.

An order, in run order 4
The same cards and lines a checkout touches, as a numbered view.

The bookshop's systems

17 cards · 18 lines · 3 boxes

Who uses the shop, what runs it, and where an order goes.



Legend **Hook / guard** fires on its own · **Client / UI** people and screens · **Service / API** runs the logic · **Agent / AI** an agent or model · **Queue / Event** hands work on · **Data store** keeps data · **Infra / Cloud** hosts, observability · **External** outside the system · **File / Doc** a document or config · **Person** someone who uses it · **Browser app** runs in a web browser · **Mobile app** runs on a phone · **Agent** an AI agent · **Command line** a CLI or a script · **Desktop app** runs in a window · **Package** a module or a folder of code · **calls** runs or invokes it · **feeds** hands work on · **triggers** suggests, offers, fires · **state** reads or writes a file, board, config · **guard** a hook steps in

Cards 17

Card	Kind	What it is
Shoppers `browse and buy`	Client / UI	The people who buy books: a `person` card is drawn as a person.
Shop staff `pack and answer`	External	People outside the system: a `user` card is drawn as a person.
Storefront `React`	Client / UI	A `frontend` card: a browser app, its title bar and address field over the card.
Shop app `iOS and Android`	Client / UI	A `mobile` card, a phone on its side, marked NEW.
Back office `staff only`	Client / UI	Its own `shape`, a window: a card's `shape` wins over its kind.

Cards 17 (continued)

Card	Kind	What it is
Admin CLI `python3`	Service / API	A `cli` card, a terminal; `noSub` keeps its sub-line in the panel only.
Shop API `FastAPI`	Service / API	The heart of the shop, drawn as a main step, with chips, a badge in a tone and an edit mark.
Pricing `discounts, tax`	Service / API	A `module` card: a folder.
Fraud check `fires on its own`	Hook / guard	A `hook` card: a hexagon.
Orders `PostgreSQL`	Data store	A `database` card: a cylinder.

The bookshop's systems (continued)

Cards 17 (continued)

Card	Kind	What it is
Order queue `RabbitMQ`	Queue / Event	A `queue` card: a pipe, open where the work leaves.
Support agent `drafts replies`	Agent / AI	An `agent` card: a robot.
returns.md `returns policy`	File / Doc	A `file` card: a page with its corner folded.
Hosts `two servers`	Infra / Cloud	A `cloud` card: a box seen in perspective.
Card gateway `cards, refunds`	External	Outside the shop: an `external` card is dashed.
Gift cards `next quarter`	Service / API	Not built yet: a PLANNED card is dashed.
Reminder bot `nudges carts`	Agent / AI	Switched off: a REMOVED card is dashed.

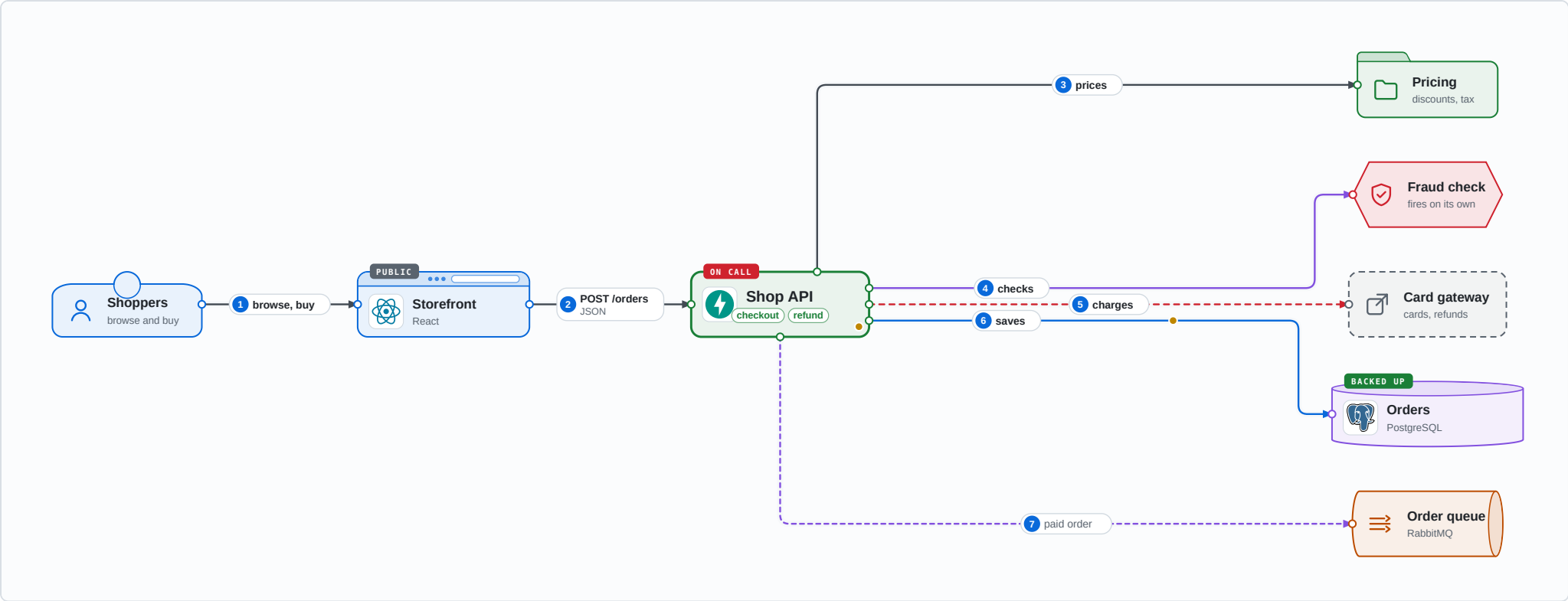
Lines 18

#	From → to	Label	Technology
1	Shoppers → Storefront	browse, buy	
2	Storefront → Shop API	POST /orders	JSON
3	Shop API → Pricing	prices	
3a	Shop API → Fraud check	checks	
4	Shop API → Card gateway	charges	
5	Shop API → Orders	saves	
6	Shop API → Order queue	paid order	
	Shoppers → Shop app	on the go	
	Shop app → Shop API	order	
	Back office → Shop API	manage	
	Admin CLI → Shop API	runs jobs	
	Order queue → Support agent	tickets	
	Support agent → returns.md	reads	
	returns.md → Card gateway	if within 14 days	
	Shop staff → Back office	pack, answer	
	Shop API → Gift cards	redeem	
	Reminder bot → Storefront	nudges	
	Hosts → Shop platform	runs	

Boxes 3

Box	Holds
Shop platform	Shop API, Pricing, Fraud check and Gift cards
Data	Orders and Order queue
Support	Support agent and returns.md

The same cards and lines a checkout touches, as a numbered view.



Legend **Hook / guard** fires on its own · **Client / UI** people and screens · **Service / API** runs the logic · **Queue / Event** hands work on · **Data store** keeps data · **External** outside the system · **Person** someone who uses it · **Browser app** runs in a web browser · **Package** a module or a folder of code · **calls** runs or invokes it · **feeds** hands work on · **triggers** suggests, offers, fires · **state** reads or writes a file, board, config

Cards 8

Card	Kind	What it is
Shoppers `browse and buy`	Client / UI	The people who buy books: a `person` card is drawn as a person.
Storefront `React`	Client / UI	A `frontend` card: a browser app, its title bar and address field over the card.
Shop API `FastAPI`	Service / API	The heart of the shop, drawn as a main step, with chips, a badge in a tone and an edit mark.

Cards 8 (continued)

Card	Kind	What it is
Pricing `discounts, tax`	Service / API	A `module` card: a folder.
Fraud check `fires on its own`	Hook / guard	A `hook` card: a hexagon.
Card gateway `cards, refunds`	External	Outside the shop: an `external` card is dashed.

Cards 8 (continued)

Card	Kind	What it is
Orders `PostgreSQL`	Data store	A `database` card: a cylinder.
Order queue `RabbitMQ`	Queue / Event	A `queue` card: a pipe, open where the work leaves.

Lines 7

#	From → to	Label	Technology
1	Shoppers → Storefront	browse, buy	
2	Storefront → Shop API	POST /orders	JSON
3	Shop API → Pricing	prices	
4	Shop API → Fraud check	checks	
5	Shop API → Card gateway	charges	
6	Shop API → Orders	saves	
7	Shop API → Order queue	paid order	