C4 diagram

2026-11-19 · `examples/types/c4.json`

6 views · 32 cards · 41 lines

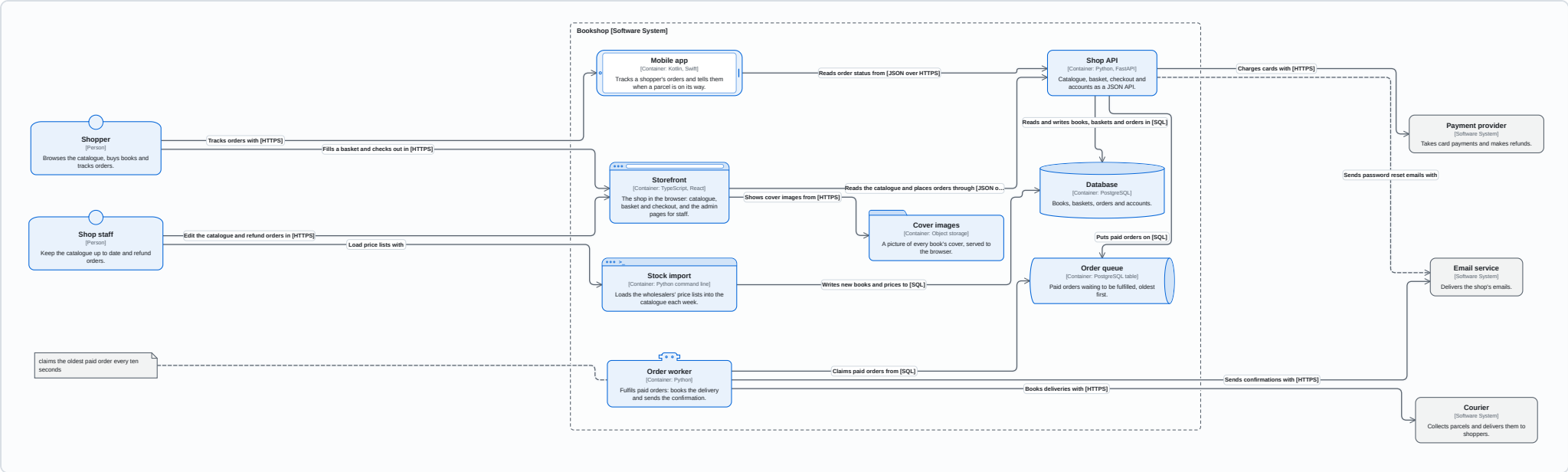
Contents

Bookshop: containers 2

- Bookshop: system context 4
- The bookshop's systems: landscape 5
- Shop API: components 7
- Placing an order 9
- Bookshop: production deployment 11

Bookshop: containers

15 cards · 16 lines



Legend **Person** a person **Container, browser** a container **Container, mobile** a container **Container** a container **Container, robot** a container
Container, cylinder a container **Container, pipe** a container **Container, folder** a container **Container, terminal** a container
External software system outside the system, drawn grey **Relationship** uses, with its technology in brackets **Implied relationship** stands for relationships between what the two hold
Comment link what a comment is about

Elements 15

Card	Kind	Members
Bookshop	Software system	Sells books online: catalogue, basket, checkout and order tracking.
Shopper	Person	Browses the catalogue, buys books and tracks orders.
Shop staff	Person	Keep the catalogue up to date and refund orders.
Storefront	Container	TypeScript, React The shop in the browser: catalogue, basket and checkout, and the admin pages for staff.
Mobile app	Container	Kotlin, Swift Tracks a shopper's orders and tells them when a parcel is on its way.

Elements 15 (continued)

Card	Kind	Members
Shop API	Container	Python, FastAPI Catalogue, basket, checkout and accounts as a JSON API.
Order worker	Container	Python Fulfils paid orders: books the delivery and sends the confirmation.
Database	Container	PostgreSQL Books, baskets, orders and accounts.
Order queue	Container	PostgreSQL table Paid orders waiting to be fulfilled, oldest first.

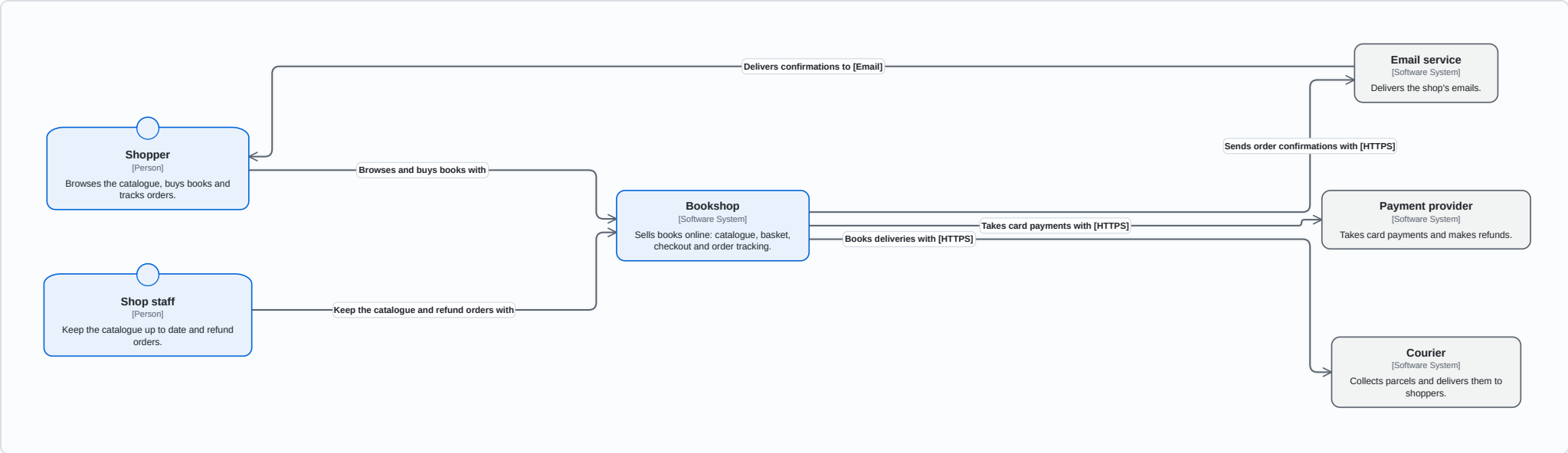
Bookshop: containers (continued)

Elements 15 (continued)

Card	Kind	Members
Cover images	Container	Object storage A picture of every book's cover, served to the browser.
Stock import	Container	Python command line Loads the wholesalers' price lists into the catalogue each week.
Payment provider	Software system	Takes card payments and makes refunds. external
Email service	Software system	Delivers the shop's emails. external
Courier	Software system	Collects parcels and delivers them to shoppers. external
claims the oldest paid order every ten seconds	Comment	

Lines 16

From → to	Label	Kind
Shopper → Storefront	Fills a basket and checks out in [HTTPS]	Relationship
Shopper → Mobile app	Tracks orders with [HTTPS]	Relationship
Shop staff → Storefront	Edit the catalogue and refund orders in [HTTPS]	Relationship
Shop staff → Stock import	Load price lists with	Relationship
Storefront → Shop API	Reads the catalogue and places orders through [JSON over HTTPS]	Relationship
Mobile app → Shop API	Reads order status from [JSON over HTTPS]	Relationship
Storefront → Cover images	Shows cover images from [HTTPS]	Relationship
Shop API → Payment provider	Charges cards with [HTTPS]	Relationship
Shop API → Database	Reads and writes books, baskets and orders in [SQL]	Relationship
Shop API → Order queue	Puts paid orders on [SQL]	Relationship
Order worker → Order queue	Claims paid orders from [SQL]	Relationship
Order worker → Courier	Books deliveries with [HTTPS]	Relationship
Order worker → Email service	Sends confirmations with [HTTPS]	Relationship
Stock import → Database	Writes new books and prices to [SQL]	Relationship
claims the oldest paid order every ten seconds → Order worker		Comment link
Shop API → Email service	Sends password reset emails with	Implied relationship



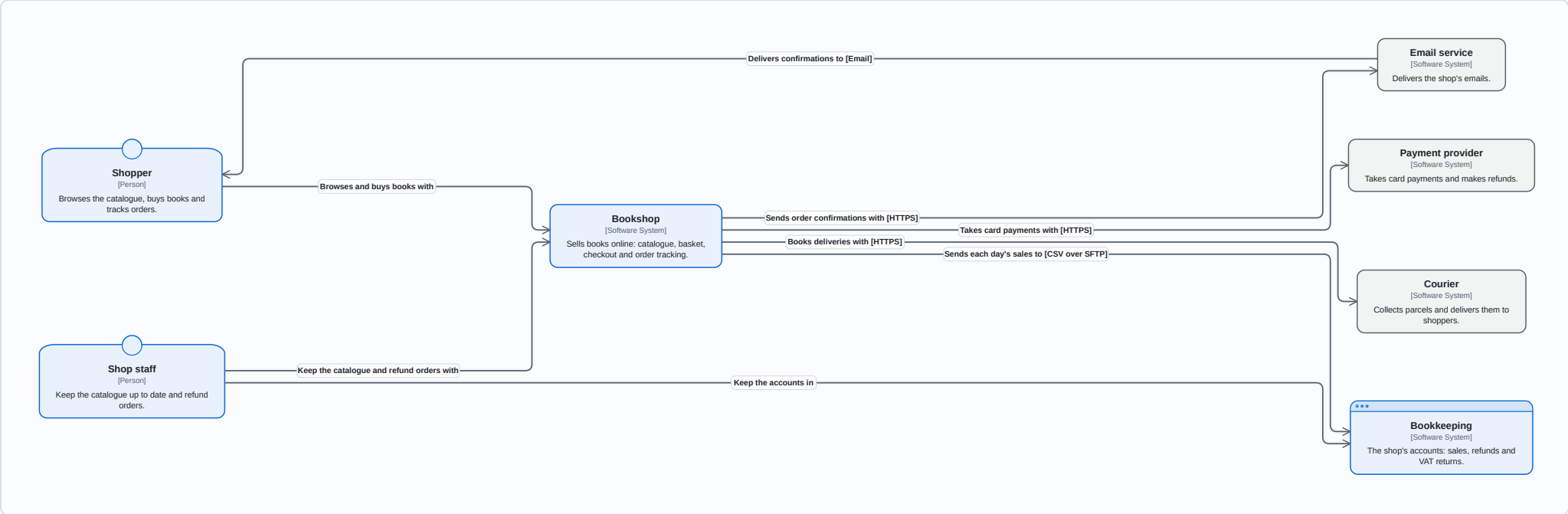
Legend **Person** a person **Software system** a software system **External software system** outside the system, drawn grey **Relationship** uses, with its technology in brackets

Elements 6

Card	Kind	Members
Shopper	Person	Browses the catalogue, buys books and tracks orders.
Shop staff	Person	Keep the catalogue up to date and refund orders.
Bookshop	Software system	Sells books online: catalogue, basket, checkout and order tracking.
Payment provider	Software system	Takes card payments and makes refunds. external
Email service	Software system	Delivers the shop's emails. external
Courier	Software system	Collects parcels and delivers them to shoppers. external

Lines 6

From → to	Label	Kind
Shopper → Bookshop	Browses and buys books with	Relationship
Shop staff → Bookshop	Keep the catalogue and refund orders with	Relationship
Bookshop → Payment provider	Takes card payments with [HTTPS]	Relationship
Bookshop → Email service	Sends order confirmations with [HTTPS]	Relationship
Bookshop → Courier	Books deliveries with [HTTPS]	Relationship
Email service → Shopper	Delivers confirmations to [Email]	Relationship



Legend **Person** a person **Software system** a software system **Software system, window** a software system **External software system** outside the system, drawn grey

Relationship uses, with its technology in brackets

Elements 7

Card	Kind	Members
Shopper	Person	Browses the catalogue, buys books and tracks orders.
Shop staff	Person	Keep the catalogue up to date and refund orders.
Bookshop	Software system	Sells books online: catalogue, basket, checkout and order tracking.
Bookkeeping	Software system	The shop's accounts: sales, refunds and VAT returns.
Payment provider	Software system	Takes card payments and makes refunds. external

Elements 7 (continued)

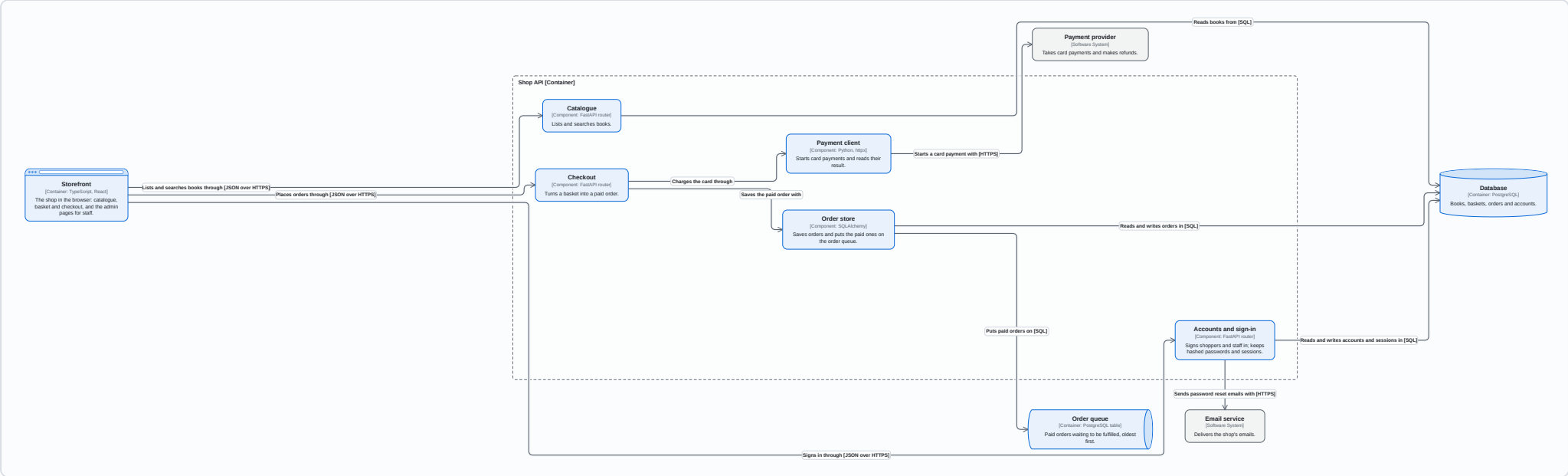
Card	Kind	Members
Email service	Software system	Delivers the shop's emails. external
Courier	Software system	Collects parcels and delivers them to shoppers. external

Lines 8

From → to	Label	Kind
Shopper → Bookshop	Browses and buys books with	Relationship
Shop staff → Bookshop	Keep the catalogue and refund orders with	Relationship
Bookshop → Payment provider	Takes card payments with [HTTPS]	Relationship

Lines 8 (continued)

From → to	Label	Kind
Bookshop → Email service	Sends order confirmations with [HTTPS]	Relationship
Bookshop → Courier	Books deliveries with [HTTPS]	Relationship
Email service → Shopper	Delivers confirmations to [Email]	Relationship
Bookshop → Bookkeeping	Sends each day's sales to [CSV over SFTP]	Relationship
Shop staff → Bookkeeping	Keep the accounts in	Relationship



Legend **Container, browser** a container **Component** a component **Container, cylinder** a container **Container, pipe** a container

External software system outside the system, drawn grey **Relationship** uses, with its technology in brackets

Elements 11

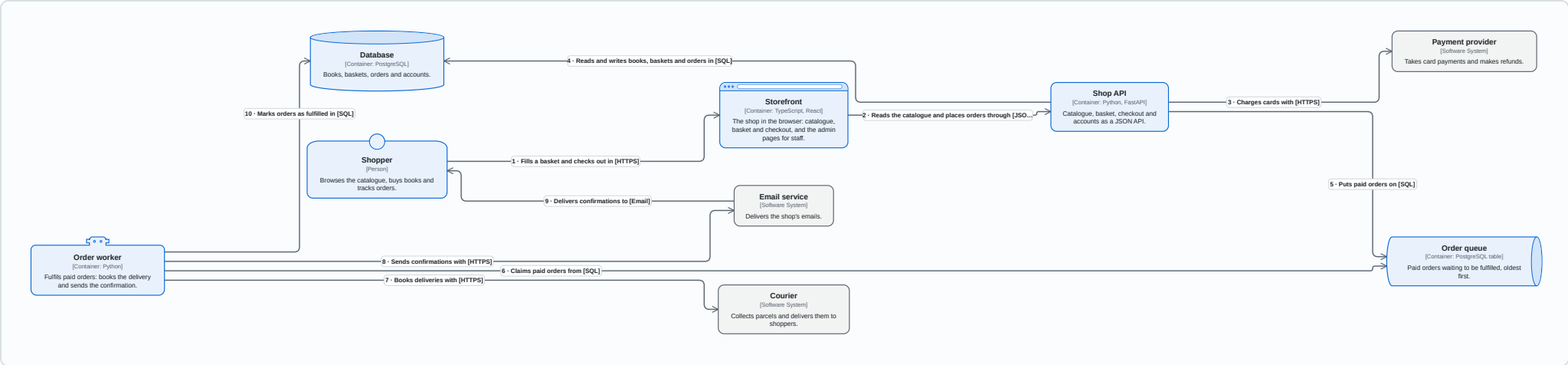
Card	Kind	Members
Shop API	Container	Python, FastAPI Catalogue, basket, checkout and accounts as a JSON API.
Storefront	Container	TypeScript, React The shop in the browser: catalogue, basket and checkout, and the admin pages for staff.
Catalogue	Component	FastAPI router Lists and searches books.
Checkout	Component	FastAPI router Turns a basket into a paid order.
Accounts and sign-in	Component	FastAPI router Signs shoppers and staff in; keeps hashed passwords and sessions.

Elements 11 (continued)

Card	Kind	Members
Payment client	Component	Python, httpx Starts card payments and reads their result.
Order store	Component	SQLAlchemy Saves orders and puts the paid ones on the order queue.
Database	Container	PostgreSQL Books, baskets, orders and accounts.
Order queue	Container	PostgreSQL table Paid orders waiting to be fulfilled, oldest first.
Payment provider	Software system	Takes card payments and makes refunds. external
Email service	Software system	Delivers the shop's emails. external

Lines 11

From → to	Label	Kind
Storefront → Catalogue	Lists and searches books through [JSON over HTTPS]	Relationship
Storefront → Checkout	Places orders through [JSON over HTTPS]	Relationship
Storefront → Accounts and sign-in	Signs in through [JSON over HTTPS]	Relationship
Checkout → Payment client	Charges the card through	Relationship
Payment client → Payment provider	Starts a card payment with [HTTPS]	Relationship
Checkout → Order store	Saves the paid order with	Relationship
Order store → Database	Reads and writes orders in [SQL]	Relationship
Order store → Order queue	Puts paid orders on [SQL]	Relationship
Catalogue → Database	Reads books from [SQL]	Relationship
Accounts and sign-in → Database	Reads and writes accounts and sessions in [SQL]	Relationship
Accounts and sign-in → Email service	Sends password reset emails with [HTTPS]	Relationship



Legend **Person** a person **Container, browser** a container **Container** a container **External software system** outside the system, drawn grey **Container, cylinder** a container **Container, pipe** a container **Container, robot** a container **Relationship** a step, numbered in time order, with its technology in brackets

Elements 9

Card	Kind	Members
Shopper	Person	Browses the catalogue, buys books and tracks orders.
Storefront	Container	TypeScript, React The shop in the browser: catalogue, basket and checkout, and the admin pages for staff.
Shop API	Container	Python, FastAPI Catalogue, basket, checkout and accounts as a JSON API.
Payment provider	Software system	Takes card payments and makes refunds. external
Database	Container	PostgreSQL Books, baskets, orders and accounts.
Order queue	Container	PostgreSQL table Paid orders waiting to be fulfilled, oldest first.
Order worker	Container	Python Fulfils paid orders: books the delivery and sends the confirmation.

Elements 9 (continued)

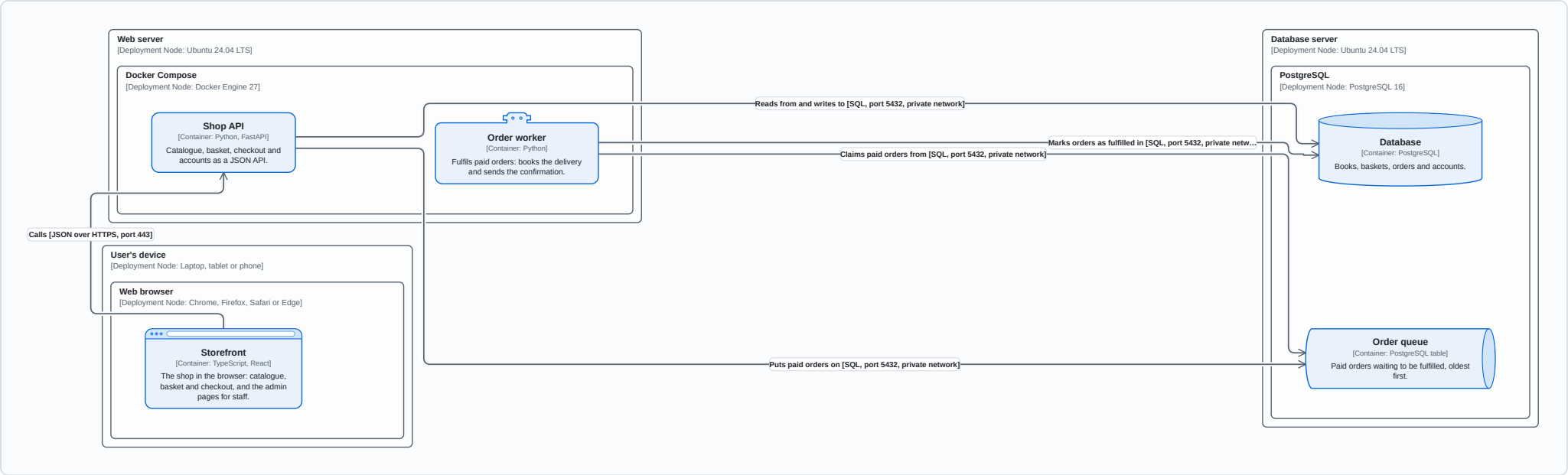
Card	Kind	Members
Courier	Software system	Collects parcels and delivers them to shoppers. external
Email service	Software system	Delivers the shop's emails. external

Lines 10

From → to	Label	Kind
Shopper → Storefront	1 · Fills a basket and checks out in [HTTPS]	Relationship
Storefront → Shop API	2 · Reads the catalogue and places orders through [JSON over HTTPS]	Relationship
Shop API → Payment provider	3 · Charges cards with [HTTPS]	Relationship
Shop API → Database	4 · Reads and writes books, baskets and orders in [SQL]	Relationship
Shop API → Order queue	5 · Puts paid orders on [SQL]	Relationship
Order worker → Order queue	6 · Claims paid orders from [SQL]	Relationship

Lines 10 (continued)

From → to	Label	Kind
Order worker → Courier	7 · Books deliveries with [HTTPS]	Relationship
Order worker → Email service	8 · Sends confirmations with [HTTPS]	Relationship
Email service → Shopper	9 · Delivers confirmations to [Email]	Relationship
Order worker → Database	10 · Marks orders as fulfilled in [SQL]	Relationship



Legend **Deployment node** a host, runtime or device; it holds what runs on it · **Container instance, browser** a container deployed here, drawn as its container · **Container instance** a container deployed here, drawn as its container · **Container instance, robot** a container deployed here, drawn as its container · **Container instance, cylinder** a container deployed here, drawn as its container · **Container instance, pipe** a container deployed here, drawn as its container · **Relationship** uses, with its technology in brackets

Elements 11

Card	Kind	Members
User's device	Deployment node	Laptop, tablet or phone A shopper's or a staff member's own device.
Web browser	Deployment node	Chrome, Firefox, Safari or Edge
Storefront	Container instance	Instance of Storefront TypeScript, React The shop in the browser: catalogue, basket and checkout, and the admin pages for staff.
Web server	Deployment node	Ubuntu 24.04 LTS The web tier's server.
Docker Compose	Deployment node	Docker Engine 27

Elements 11 (continued)

Card	Kind	Members
Shop API	Container instance	Instance of Shop API Python, FastAPI Catalogue, basket, checkout and accounts as a JSON API.
Order worker	Container instance	Instance of Order worker Python Fulfils paid orders: books the delivery and sends the confirmation.
Database server	Deployment node	Ubuntu 24.04 LTS Backed up every night.
PostgreSQL	Deployment node	PostgreSQL 16

Elements 11 (continued)

Card	Kind	Members
Database	Container instance	Instance of Database PostgreSQL Books, baskets, orders and accounts.
Order queue	Container instance	Instance of Order queue PostgreSQL table Paid orders waiting to be fulfilled, oldest first.

Lines 5

From → to	Label	Kind
Storefront → Shop API	Calls [JSON over HTTPS, port 443]	Relationship
Shop API → Database	Reads from and writes to [SQL, port 5432, private network]	Relationship
Shop API → Order queue	Puts paid orders on [SQL, port 5432, private network]	Relationship
Order worker → Order queue	Claims paid orders from [SQL, port 5432, private network]	Relationship
Order worker → Database	Marks orders as fulfilled in [SQL, port 5432, private network]	Relationship