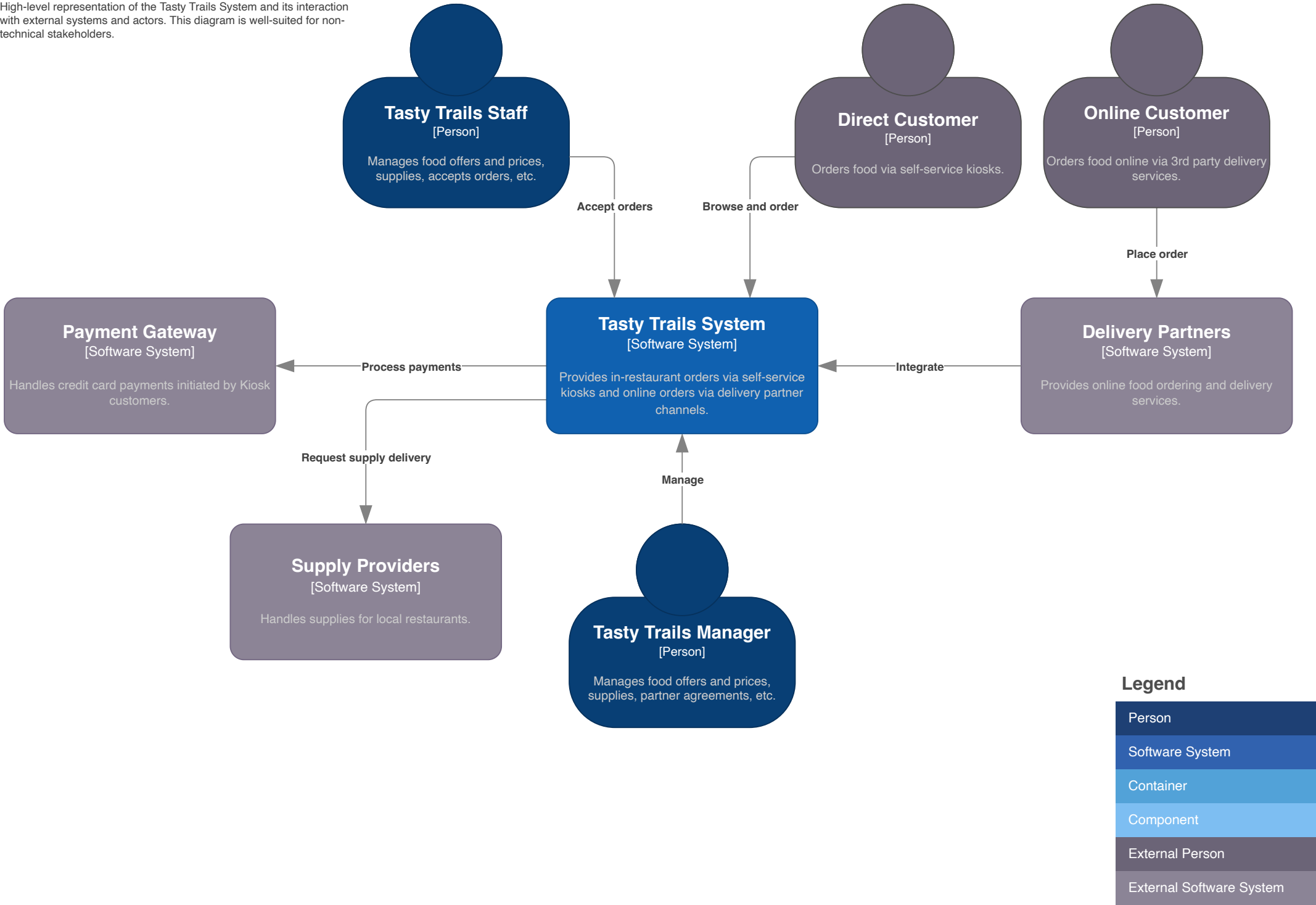# Tasty Trails System Design

[System Context] Tasty Trails C1 Diagram

High-level representation of the Tasty Trails System and its interaction with external systems and actors. This diagram is well-suited for non-technical stakeholders.

Tasty Trails Staff
[Person]

Manages food offers and prices, supplies, accepts orders, etc.

Direct Customer
[Person]

Orders food via self-service kiosks.

Online Customer
[Person]

Orders food online via 3rd party delivery services.

Accept orders

Browse and order

Place order

Payment Gateway
[Software System]

Handles credit card payments initiated by Kiosk customers.

Tasty Trails System
[Software System]

Provides in-restaurant orders via self-service kiosks and online orders via delivery partner channels.

Delivery Partners
[Software System]

Provides online food ordering and delivery services.

Process payments

Integrate

Request supply delivery

Manage

Supply Providers
[Software System]

Handles supplies for local restaurants.

Tasty Trails Manager
[Person]

Manages food offers and prices, supplies, partner agreements, etc.

Legend

Person
Software System
Container
Component
External Person
External Software System

[Containers] Tasty Trails C2 Diagram

Deep dive into the Tasty Trails software system, this diagram
depicts major components and their relationships, as well as
sample technologies and communication protocols. This diagram
is primarily intended for technical stakeholders.

Online Customer
[Person]

Orders food online via 3rd party delivery services.

Direct Customer
[Person]

Orders food via self-service kiosks.

Restaurant Staff
[Person]

Manages food offers and prices, supplies, accepts orders, etc.

Tasty Trails Manager
[Person]

Manages food offers and prices, supplies, partner agreements, etc.

Delivery Partners
[Software System]

Provides online food ordering and delivery services.

Place orders

Notify order
status change
[SMS]

Notify order
status change
[SMTP]

Manage

Place orders

Track orders

Send order
requests
[HTTPS]

Tasty Trails
[Software System]

[Local Network]

Self-service Kiosk UI
[Container: Blazor]

Description of web browser container role/responsibility.

Kitchen Display
[Container: Blazor]

Displays customer order information to the restaurant staff.

Notification API
[Container: ASP.NET]

Notifies customers and restaurant staff about order status changes.

Central Management Portal
[Container: Blazor]

Interface for managing offer and prices, partner agreements, inventory and customer data.

Notify
[HTTPS]

Handle orders
[HTTPS]

Handle orders
[HTTPS]

API Gateway
[Container: Nginx]

Manages customer authentication and routing to required services.

Subscribe to order placements
[AMQPS]

Publish order requests
[TCP 443]

Delivery Channel API
[Container: ASP.NET]

Exposes APIs for integration to delivery partners and manages orders placed via delivery channels.

Route order requests
[HTTPS]

Manage offer, prices and partner agreements
[HTTPS]

Subscribe to price and offer changes
[AMQPS]

Synchronization Service
[Container: Azure Service Bus]

Main synchronization service between restaurant nodes and the central system.

Subscribe to order requests
[TCP 443]

Route order history and tracking requests
[HTTPS]

API Gateway
[Container: Azure Application Gateway]

Route order requests
[HTTPS]

Publish and subscribe to order requests
[AMQPS]

Order History API
[Container: ASP.NET]

Manages order history for registered customers.

Route price and offer change requests
[HTTPS]

Route customer info requests
[HTTPS]

Route partner agreement requests
[HTTPS]

Route inventory requests
[HTTPS]

Order Management API
[Container: ASP.NET]

Collects customer order requests and stores them locally for business continuity. Synchronizes order requests with the central system and subscribes to online order requests.

Get offer and prices
[HTTPS]

SSK Management API
[Container: ASP.NET]

Manages offer and prices, inventory and supplies at a restaurant location.

Publish offer and price changes
[AMQPS]

Read/write
[TCP 443]

Read/write
[TCP 443]

Payment Gateway
[Software System]

Handles credit card payments initiated by Kiosk customers.

Handle payments
[HTTPS]

Read/write
[TCP 443]

Read/write
[TCP 443]

Read
[TCP 443]

Orders DB
[Container: PostgreSQL]

Stores order requests.

Local Management DB
[Container: PostgreSQL]

Stores latest offer and prices, inventory and supply levels.

Order History DB
[Container: Azure SQL]

Stores order history for registered customers.

Price and Offer API
[Container: ASP.NET]

Central service for managing offer and prices that are then propagated to all restaurants.

Customer API
[Container: ASP.NET]

Manages customer profiles.

Partner Agreement API
[Container: ASP.NET]

Manages partner agreements.

Inventory API
[Container: ASP.NET]

Central service for inventory management and supply deliveries.

Request supply delivery
[HTTPS]

Supply Providers
[Software System]

Handles supplies for local restaurants.

Subscribe to supply delivery requests
[AMQPS]

Cache
[Container: Redis]

Caches offer, prices, and inventory levels; TTL is up to 24 hours.

Check supply levels
[TCP 443]

Scheduler Service
[Container: Quartz.NET]

Monitors the supply levels and periodically requests supply delivery.

Publish supply delivery requests
[TCP 443]

Price and Offer Dispatcher
[Container: Time Trigger Azure Function]

Scheduled service for updating restaurant devices with latest offer and prices.

Read/write
[TCP 443]

Read/write
[TCP 443]

Price and Offer DB
[Container: Azure SQL]

Read/write
[TCP 443]

Customer DB
[Container: Azure SQL]

Read/write
[TCP 443]

Partner Agreement DB
[Container: Blob Storage]

Read/write
[TCP 443]

Inventory DB
[Container: Azure SQL]

[Central System]

Legend

Person
Software System
Container
Component
External Person
External Software System