

Lemmid Store koppelen met keukens

Integratie met externe systemen die u niet beheert

Willem L. Middelkoop

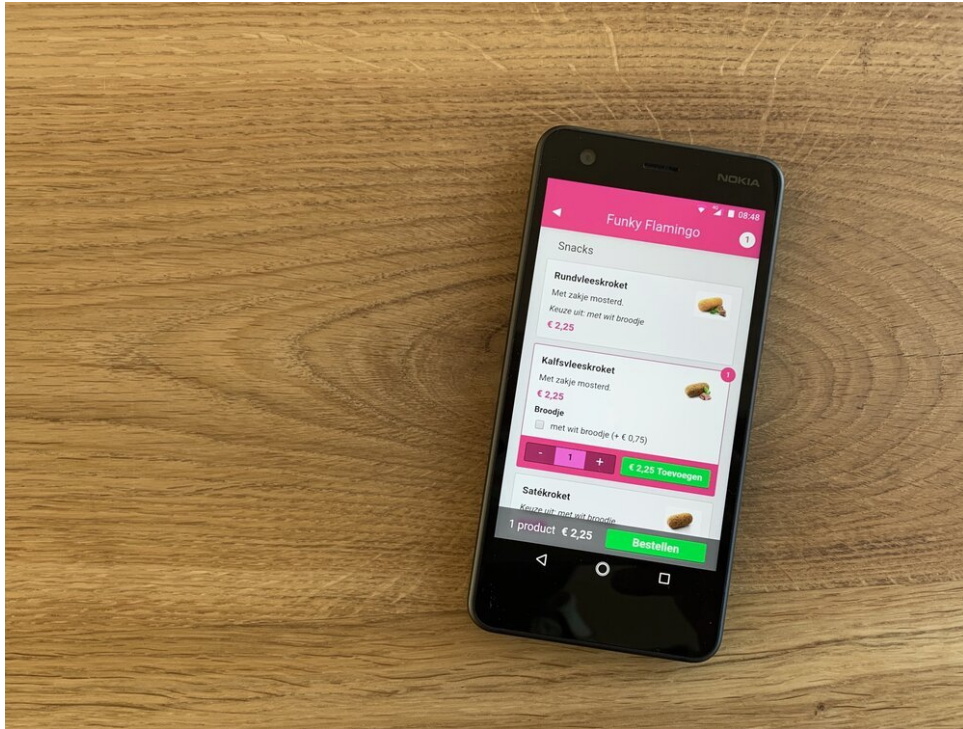
12 juni 2020

Stks	Stks	Omschrijving	Prijs	Ttl
1		Familiezak friet	15.00	15.00
	1	6 personen	0.00	0.00
4		Kipnuggets	3.75	15.00
1		Satékroket	2.25	2.25

Als onderdeel van de food-bestel-app die ik aan het bouwen ben, moest ik een betrouwbare manier ontwerpen om de app te koppelen aan externe systemen. Deze externe systemen vallen buiten mijn directe controle en omvatten verschillende kassasystemen, keukenmanagementsystemen en ticketprinters. Lees verder voor meer informatie over ontwerpen voor het onbekende en onbetrouwbare.

Bestel-app voor eten

De app die ik bouw is [Lemmid Store](#), het is een gebruiksvriendelijke app waarmee mensen bestellingen voor afhaalmaaltijden kunnen plaatsen (en betalen). Je kunt alle [berichten op Lemmid vinden](#).



Lemmid Store - een app om afhaalmaaltijden te bestellen

Externe systemen

De noodzaak om verbinding te maken met externe systemen wordt gedreven door de wens om de bestel-app voor eten te integreren in de bedrijfsprocessen die verband houden met de bezorgservice voor eten. Denk aan zaken als:

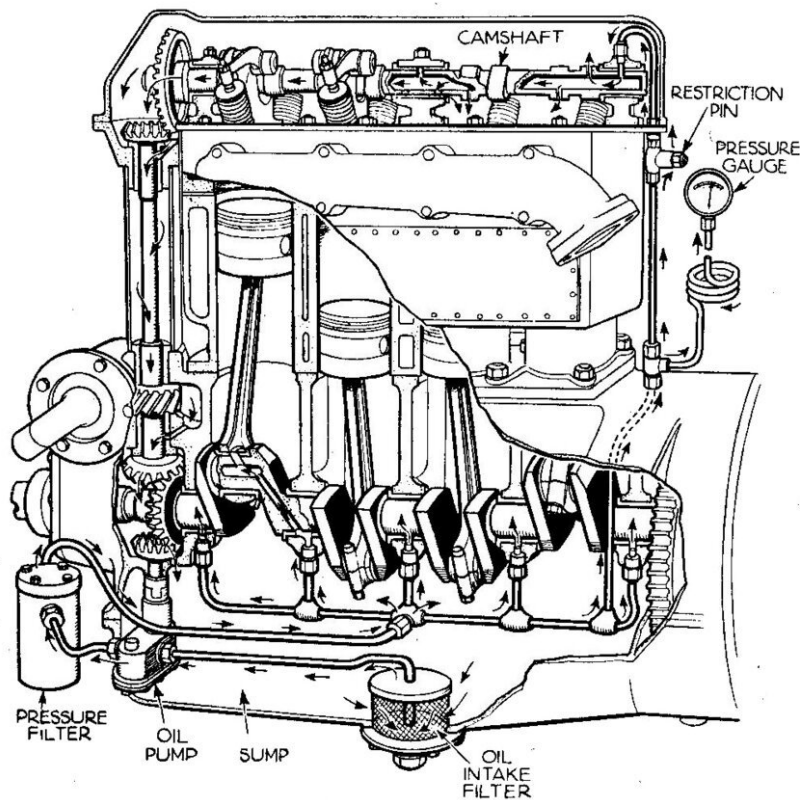
- het verzamelen van bestelgegevens in administratieve gegevens (bijv. boekhouding)
- het doorsturen van bestellingen naar de keukens, waar de maaltijden worden bereid en verpakt
- het bieden van gemakkelijke toegang tot route-informatie, die laat zien waar het eten moet worden bezorgd
- het inplannen van voedselbestellingen die bedoeld zijn voor latere bezorging

Afhalen				
[Barcode]				
Ref: BEA5EA				
Besteld: 16:35		Leveren: 2020-06-12 17:30:42		
Stks	Stks	Omschrijving	Prijs	Ttl
1		Familiezak friet	15.00	15.00
	1 6 personen		0.00	0.00
4		Kipnuggets	3.75	15.00
1		Satékroket	2.25	2.25

Geprinte ticket - via een externe verbinding - inclusief gedetailleerde bestelgegevens zoals product modifiers (keuzes van de klant per product)

Uitdagingen met het verbinden met externe systemen

Je realiseert het je misschien niet, maar verbinding maken met externe systemen kan lastig zijn. Als je de bestel-app als een soort "motor" beschouwt (zoals in je auto), wil je deze motor in wezen integreren met externe onderdelen. Zoals je je kunt voorstellen, zijn niet alle onderdelen even geschikt om onder de figuurlijke motorkap te passen. Om het nog uitdagender te maken: je weet niet zeker hoe betrouwbaar de externe onderdelen zijn. Je moet je integratie zo ontwerpen dat verschillende externe onderdelen gemakkelijk kunnen worden aangesloten en dat de prestaties onafhankelijk zijn van de betrouwbaarheid van deze externe onderdelen.

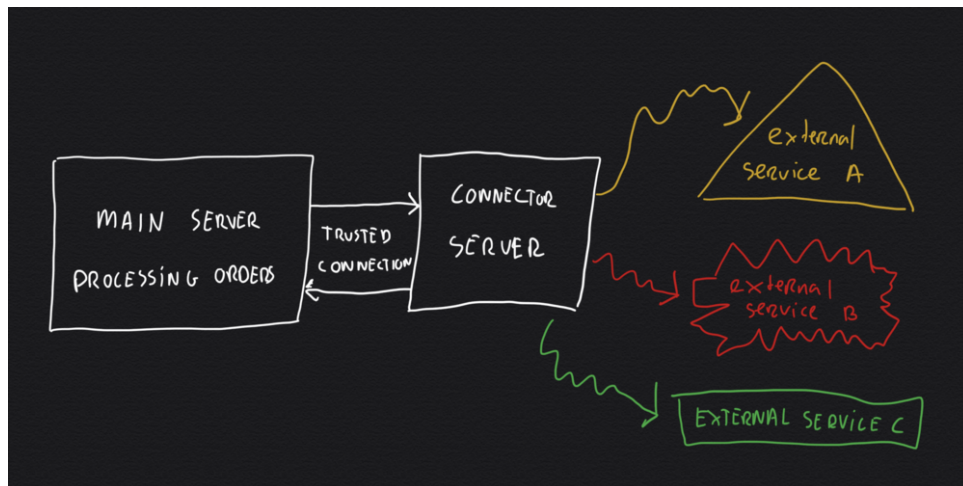


Het integreren van externe systemen in je app is als het monteren van onderdelen van derden op een motor: je moet zorgvuldig rekening houden met de passing en betrouwbaarheid (image)

Net als bij een verbrandingsmotor moet je rekening houden met de pasvorm en betrouwbaarheid wanneer je je software integreert met externe systemen. **Je kunt er niet van uitgaan dat externe software:**

- perfect past
- altijd consistent is
- altijd beschikbaar/aanwezig is
- betrouwbaar is

Om te ontwerpen voor deze nogal grimmige kijk op externe verbindingen, vond ik het een goed idee om de belangrijkste backendserver van mijn bestel-app voor eten de verantwoordelijkheid voor het verbinden met externe systemen over te laten aan een speciaal gebouwde 'connectorserver'. De connectorserver zou dan onafhankelijk verbinding maken met het externe systeem. Op deze manier communiceert de hoofdservr alleen met 'first party' onderdelen en systemen, waardoor de risico's van misfits en mogelijke downtime worden geminimaliseerd.



Het scheiden van de belangrijkste backend server van de externe systemen door het introduceren van een speciaal gebouwde 'connector server'

In de praktijk betekent dit dat de 'connectorserver' eigenlijk een andere machine is, met een eigen softwarestack. Er zijn een paar belangrijke voordelen aan deze segmentatie:

- **beveiliging:** de connectorserver bewaart geen volledig overzicht van alle bestellingen, alleen de bestellingen die moeten worden doorgestuurd naar een extern systeem worden tijdelijk in het geheugen bewaard. Een mogelijk beveiligingsprobleem in een extern systeem zal slechts een minimale hoeveelheid gegevens blootstellen.
- **past overall:** de connectorserver kan vele verschillende protocollen en manieren gebruiken om verbinding te maken met externe systemen. Verbindingen met externe systemen kunnen worden toegevoegd, net zoals de server 'een nieuwe taal leert spreken'. Omdat het al 'meertalig' is, is het toevoegen van een nieuw dialect of taal vaak eenvoudig. Verbinding maken met verschillende externe API's is gemakkelijker wanneer je alle bouwstenen hebt om gegevens te (her)structureren.
- **omgaan met onbetrouwbaarheid:** Een eenvoudig wachtrijmechanisme in de connectorserver stelt het in staat om elegant om te gaan met mogelijke downtime en problemen met externe verbindingen. Door per externe service te loggen en in de wachtrij te plaatsen, wordt het mogelijk om de impact op andere onderdelen en verbindingen te minimaliseren wanneer één service uitvalt.

Conclusie

Hoe verleidelijk het ook is om gewoon nieuwe en mooie onderdelen aan je app vast te schroeven, je moet je setup zorgvuldig overwegen om ervoor te zorgen dat deze betrouwbaar presteert.

Het ontwerpen van back-endserversoftware om verbinding te maken met externe API's is een uitdaging. Je moet hopen op het beste, maar ontwerpen voor het ergste!