

Kitchen Connect



From customer discovery
to connected cooking

A customer-led approach to qualification, risk,
integration design and a demonstrable outcome.

01 DISCOVER

Understand the need

02 DE-RISK

Decide what to prove

03 DEMONSTRATE

Show the whole journey

Start with a customer decision—not a feature list.

Hypothetical brief: an appliance brand wants a useful reason for customers to return to its connected oven.

**“Turn a dinner idea into a prepared cooking session—
using the pantry and filling the gaps.”**

CONSUMER

Less planning friction

Use what is available; understand what is missing.

APPLIANCE BRAND

Repeat cooking use

A proposed outcome to validate—not a proven uplift.

DELIVERY TEAM

A bounded integration

Clarify responsibilities before promising a live rollout.

Context [1–3]: Fresco describes connected cooking and modular integration. This customer brief and grocery extension are hypothetical.

Run discovery as a working conversation.

Capture answers, evidence, owners and unknowns—not just a list of requested features.

CONVERSATION	QUESTIONS TO ASK	LEAVE WITH
Business outcome	What is painful today? Why act now? What would change?	Problem statement + baseline
Household journey	Where do cooks abandon the task? What do they do instead?	Current journey + priority friction
Technical estate	Which device models, apps, identities and data sources exist?	System map + named technical owners
Commercial decision	Who sponsors, funds and approves this? What is the process?	Decision path + budget route
Delivery & support	Who supplies access, validates behavior and owns incidents?	Dependencies + acceptance owners
Reflect it back: “Have we understood the problem, constraints and decision correctly?”		

Qualify the opportunity before committing engineering.

Technical enthusiasm is not the same as a viable customer engagement.

COMMERCIAL

Is there a decision?

Named sponsor and buyer
Credible funding route
Agreed decision date

PLATFORM FIT

Should we solve it?

Fits platform capabilities
Bounded custom work
Build vs partner understood

DELIVERY READINESS

Can we prove it?

Customer people and data
Named dependency owners
Bounded scope and exit

PROCEED

Evidence supports a bounded POC.

CONDITIONAL

Proceed only with owned conditions.

HOLD / RE-SCOPE

Critical fit, access or ownership missing.

Make the risks explicit—and give each one an owner.

Initial risk hypotheses. Likelihood and severity require customer evidence; they are not scored here.

RISK	WHY IT MATTERS	EVIDENCE / MITIGATION	ACCOUNTABLE OWNER
No funded decision	Engineering without a next step	Sponsor, decision path and agreed POC exit	Sales + customer sponsor
Provider access / rights	Live integration may be blocked	Authorised contracts, sandbox and usage permissions	Partner / technical owner
Inventory / catalogue errors	Wrong quantities or purchase packs	Canonical units, pack rules, receipt reconciliation	Customer data owner
Device behavior / safety	Unsupported or unsafe operation	Model-level capabilities, approvals and offline fallback	Device engineering owner
Data / operational ownership	Unclear controls or incident response	Data map, retention choices and support responsibilities	Security + operations

Mocking a dependency helps the POC. It does not remove that dependency from the pilot.

Agree on what this POC can actually prove.

One recipe • 2 or 4 servings • one pantry • one substitution • one retailer • one oven

WORKING LOGIC TO BUILD

A coherent journey

Quantity and pack calculations
Review and approval controls
Delivery and cooking states
Reset and visible activity

SIMULATED BOUNDARIES

No external effects

Editable sample inventory
Fictional prices and orders
Fictional oven capabilities
In-browser provider adapters

NOT PROVEN BY THIS POC

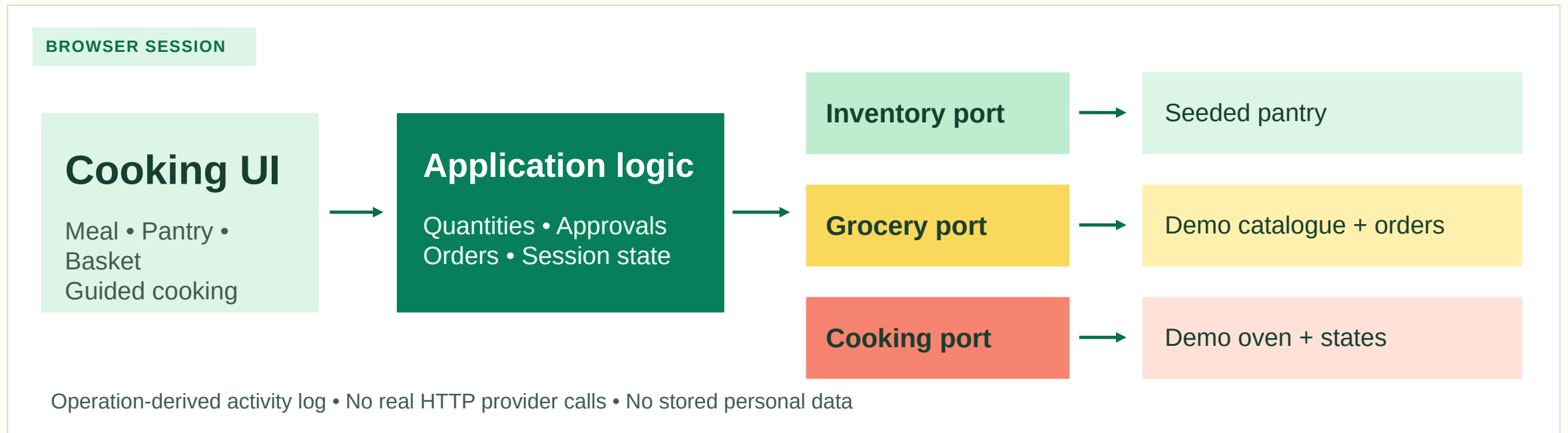
A separate pilot gate

Real Fresco / retailer access
Production scale or security
Safe physical device control
Customer adoption or ROI

Agree the acceptance tests before coding. Current implementation and test status: NOT STARTED.

Keep the boundaries visible.

Proposed POC architecture: all business operations stay inside the visitor's browser session.



LIVE PILOT: authenticated integration service + authorised provider contracts + device-specific validation

Context [2]: Fresco describes SDK/API/cloud options. The contracts shown here are our illustrative interfaces, not Fresco API definitions.

Choose dinner. Show the actual ingredient gap.

Seeded scenario: sheet-pan chicken fajitas for two. All UI below is an editable illustration.

Kitchen Connect

ILLUSTRATIVE UI — NOT A RUNNING APPLICATION

Sheet-pan chicken fajitas

2 SERVINGS

5 SHORTAGES

INGREDIENT	NEED	HAVE	MISSING
Chicken	300 g	400 g	—
Bell pepper	2 each	1 each	1 each
Tortillas	4 each	0 each	4 each

Also missing: seasoning, lime and sour cream.

What to show

Inventory is editable.

Shortages come from quantities —not a preset answer.

Partial stock matters: one more pepper is needed.

Use an alternative. Recalculate the basket.

The substitution is explicit and curated: Greek yogurt replaces sour cream for the topping. Both contain dairy.

Kitchen Connect

ILLUSTRATIVE UI — NOT A RUNNING APPLICATION

Use 60 g Greek yogurt from the pantry

PRODUCT	PACK SIZE	PACKS	SUBTOTAL
Bell peppers	3 each	1	€2.49
Tortillas	8 each	1	€1.99
Seasoning	30 g	1	€0.99
Limes	3 each	1	€1.29
Illustrative product subtotal			€6.76

5 → 4
ingredient lines

€8.25 → €6.76

Then change to four servings:
5 lines • €13.74
Seasoning rounds to two packs.

Approval, order and delivery are different events.

The visitor reviews the current basket before any simulated order is created.

1 / REVIEW

Confirm the plan

2 servings • yogurt topping
4 product lines • €6.76
Plan edits require review.

2 / ORDER

Place simulated order

SIM-... receipt created
Repeat click: same order
Pantry remains unchanged.

3 / DELIVERY

Receive purchased packs

Explicit delivery action
Receive full packs once
Then ingredients are available.

Important distinction Order confirmed ≠ ingredients delivered ≠ permission to start cooking.

Connect the cooking journey—not just an oven icon.

The program is prepared only after checking the sample device capabilities and the available ingredients.

KC-001: CORE JOURNEY

Demo Oven

Program supported • Pantry sufficient • User confirms

READY

PREHEATING

COOKING

COMPLETE

Manually advanced simulation. No physical device commands.

KC-002: FAILURE DEMO

Now take the oven offline.

Preserve the meal and receipt.
Explain the blocked action.
Offer non-connected guidance.

These are planned states. Simulated program values do not establish safe cooking instructions.

Turn demonstration claims into observable tests.

Status at this stage: NOT RUN. Pass labels require implementation evidence and review.

CLAIM TO VERIFY	OBSERVABLE ACCEPTANCE RESULT	EVIDENCE REQUIRED
The arithmetic is real	€8.25 baseline → €6.76 substituted → €13.74 for four	Domain tests + browser journey
Approval is current	Plan edits invalidate an earlier basket review	State test + UI demonstration
Retries are harmless	Repeated submit/delivery/start has no duplicate effect	Adapter + lifecycle tests
Device failure is contained	No simulated start; meal and receipt remain available	Negative tests + KC-002 demo
It is a safe-to-share demo	No real provider calls, secrets or personal data	Network review + repository scan

The interview evidence: working app + test results + implementation report + actual screenshots.

A mock POC earns a pilot decision—not a rollout.

Make the next step a separately bounded engagement with customer participation.

POC EXIT

Accept the demonstration

Agreed journey works
Evidence reviewed
Unknowns and limitations recorded

LIVE PILOT ENTRY

Validate the dependencies

Sponsor and funding path
Authorised API/device access
Security, support and safety owners

PILOT DECISION

Measure customer value

Baseline and cohort agreed
Completion/repeat-use metrics
Operational and commercial review

Agree metric definitions and targets with the customer. Do not infer ROI from a simulated journey.

Finish with a decision and named next steps.

The output is shared understanding—not “we showed some interesting technology.”

CONFIRM

Is this the problem and journey worth solving?

RESOLVE

Which unknowns prevent a real pilot, and who owns them?

COMMIT

What evidence, contribution and decision date do we agree?

Planned handover: demo link • GitHub repository • evidence report • customer/pilot decision record

Architect together. Codex builds. We review the evidence.

Keep the customer decisions with us and the bounded engineering work with the implementation agent.

OWNER	RESPONSIBILITY	OUTPUT
You	Customer story, priorities, product judgment and final acceptance	Agreed scope + human sign-off
You + ChatGPT	Architecture, risk, interfaces, acceptance tests and slice review	Bounded Codex-ready slice
Codex	Implementation, tests, screenshots, self-review and measured build record	Code + implementation report
You + ChatGPT	Validate behavior and evidence; accept or issue a focused repair	Next slice only after review

No “it runs, therefore it passed.” No next slice until the current evidence is reviewed.

Method context [4]: OpenAI recommends explicit success criteria, reusable repository guidance, testing and review.

Three slices. One complete story. No platform project.

Only KC-001 is specified for execution now. Later slices are drafted against accepted code.

SLICE	GOAL	INCLUDE	EXIT EVIDENCE
KC-001	Complete thin journey	Pantry → substitution → basket → order → delivery → oven	Exact fixtures; desktop/mobile tests; report
KC-002	Failure and presentation polish	Offline/capability controls, fallback, accessibility and browser hardening	Negative journeys; visual review; regression tests
KC-003	Publish and package	Authorised GitHub/hosting release; website link; replace slide mockups	Verified URL, clean launch, screenshots, release record

Deferred: accounts, payments, live AI, real retailer/device control, production infrastructure and multi-brand scale.

Use public context. Label our assumptions.

Sources checked 23 September 2026. Product descriptions do not establish a usable integration contract.

[1] Fresco — KitchenOS

Public platform positioning

[2] Fresco — Technical & Security

Managed and modular SDK/API/cloud integration options

[3] Fresco — AI Cooking Companion

Recipe, guidance and appliance experience descriptions

[4] OpenAI — Codex best practices

Task context, repository guidance, verification and review

Evidence boundary

Fictional customer • Illustrative UI and prices • No verified provider contracts
No live app, test pass, commercial result or deployment claimed in this blueprint