

Super Architecture Data

Third Discussion Format

Document type	Product-surface planning (shared product hosts + post-login tenant)
Status	Discussion / planning baseline (pre-tenant.product host model)
Frontends	2 codebases
Products	3 (POS, Landscaping/Service, E-Commerce) — sometimes sold as Superset

1. Business Intent (Your Plan)

Product Surfaces

Product surface	Example URL	Who uses it	What they see
POS	pos.nexzentek.com	Staff	Only POS/nursery modules + admin retail ops
Landscaping	landscaping.nexzentek.com	Staff	Only service/landscaping modules
E-Commerce	ecom.nexzentek.com	Customers (+ related admin links from POS)	Storefront web
Superset (bundle)	egf.nexzentek.com	Staff	POS + Landscaping together
E-Com in superset	still ecom.nexzentek.com	Customers of that company	Same e-com frontend, company-scoped data

Two Frontends, Three Products

Frontend codebase	Serves
Frontend A — Admin/Staff app	pos.*, landscaping.*, egf.* (module visibility by host + plan)
Frontend B — E-Commerce app	ecom.* (customer storefront)

Product Packaging

Sell as	Staff access	Customer access
POS only	pos.nexzentek.com	optional ecom if plan includes it
Landscaping only	landscaping.nexzentek.com	usually no retail e-com
E-Com add-on	Admin ops via POS host	ecom.nexzentek.com
Superset (e.g. EGF)	egf.nexzentek.com (POS + Landscaping)	ecom.nexzentek.com

2. Third Discussion Routing Model

pos.nexzentek.com	----->	Staff FE	-->	show POS modules only
landscaping.nexzentek.com	----->	Staff FE	-->	show Landscaping modules only
egf.nexzentek.com	----->	Staff FE	-->	show POS + Landscaping (superset)
ecom.nexzentek.com	----->	Ecom FE	-->	customer storefront

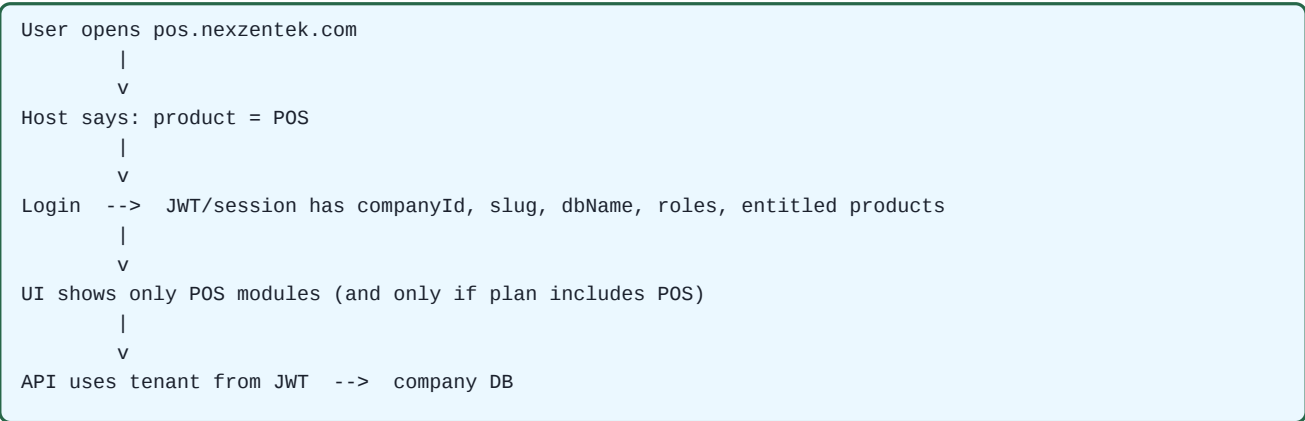
Resolution Order

1	Host → Product surface	Which modules / which frontend
2	Login / session / path / domain → Tenant	Company resolution
3	API opens that company's database	Data isolation

Key principle: Tenant is **not** in the primary product host. Product host only decides **which product experience**.

3. Staff Apps (POS / Landscaping / Superset)

How Access Works



Same pattern for: **landscaping.nexzentek.com** --> service modules only; **egf.nexzentek.com** --> POS + Landscaping modules.

Admin work on POS host

Even when e-com exists, staff still use **POS host** for: Product creation, Preorder creation, Dashboard, Sales check, Online orders check. POS host = retail + e-com **admin operations**. Ecom host = customer buying experience.

Data Source

Surface	Tenant source	DB
POS	JWT/session after login	That company DB
Landscaping	JWT/session after login	That company DB
EGF / Superset	JWT/session after login	That company DB

Pros for staff surfaces	Cons
• Clean product marketing URLs • One staff frontend with host-based module filters • Strong fit for login-first workflows • Superset is just another host profile (module pack)	• Multi-company users need workspace/company picker after login • Browser URL does not show company name • Must enforce product entitlement strictly (POS user must not open landscaping modules unless entitled)

4. E-Commerce — Main Confusion and Options

The Problem

Customer opens **ecom.nexzentek.com** **without login** and must see **one company's products**. Host alone only says "this is e-com", **not** "which company". So e-com needs a **second signal** for tenant.

Option E1 — Path-based company slug

Example: `ecom.nexzentek.com/acme/...` or `ecom.nexzentek.com/nursery/acme/...`

How	First path segment = company slug → load that company DB
Pros	Works before login; shareable links; one e-com deploy; simple DNS; good for local testing
Cons	Company visible in path; weaker "premium brand" feel; caching/routing careful; not tenant-subdomain style
Fit	Strong with third discussion

Option E2 — Custom domain per company (white label)

Example: `www.acmefarm.com` → mapped to company Acme → same e-com frontend

How	Host header maps to company; load Acme DB
Pros	Best branding; no company slug in path; works before login; excellent for paid white-label
Cons	DNS + SSL per customer; slower onboarding; more ops
Fit	Strong as premium add-on

Option E3 — Store picker on shared e-com home

Example: `ecom.nexzentek.com` → choose "Acme" → session stores tenant → browse

How	User selects company; session / X-Tenant-Slug set; then catalog loads
Pros	Simplest shared URL; no path/DNS complexity
Cons	Bad UX for retail; wrong-store risk; weak SEO; not direct deep links to a store
Fit	Weak for production retail

Option E4 — Query / invite token

Example: `ecom.nexzentek.com?company=acme` or signed token link

How	Link carries tenant; app stores in session
Pros	Easy campaigns / SMS / email links
Cons	Easy to tamper if unsigned; ugly URLs; not primary architecture
Fit	Secondary only

Option E5 — Sub-path store + optional custom domain (hybrid) ★ PREFERRED

Default: `ecom.nexzentek.com/acme` | Premium: `www.acmefarm.com` → Acme

How	Path for standard tenants; custom domain for white-label
Pros	Instant onboarding + optional branding; covers anonymous browse
Cons	Two resolution paths to maintain
Fit	Best overall for this model

5. E-com Options Comparison

Option	Anonymous catalog	Per-company data	Ops effort	SEO / share links	Recommended?
E1 Path slug	Yes	Yes	Low	Medium	Yes (default)
E2 Custom domain	Yes	Yes	High	High	Yes (add-on)
E3 Store picker	After pick	Yes	Low	Low	No (primary)
E4 Query/token	Yes	Yes	Low	Low	Support only

E5 Hybrid E1+E2	Yes	Yes	Medium	High	PREFERRED
-----------------	-----	-----	--------	------	-----------

6. End-to-End Data Flow (Third Discussion)

Staff — POS only company

pos.nexzentek.com --> login --> Company A JWT --> DB[A]
Modules: POS admin, products, sales, e-com order management (if entitled)

Staff — Landscaping only company

landscaping.nexzentek.com --> login --> Company B JWT --> DB[B]
Modules: enquiries, quotes, jobs, service invoices

Staff — Superset company (EGF style)

egf.nexzentek.com --> login --> Company C JWT --> DB[C]
Modules: POS + Landscaping
Customer shop: ecom.nexzentek.com/acme (or custom domain) --> DB[C]

Customer — anonymous browse

ecom.nexzentek.com/acme/home
|
v
Path tenant = acme
|
v
API loads Company Acme DB products
|
v
Cart/checkout writes orders into Acme DB
|
v
Visible to Acme staff on pos.nexzentek.com (orders screen)

7. Implementation Planning (Third Discussion)

Phase T1 — Product host matrix

- Define hosts: pos, landscaping, egf, ecom
- Map each host → allowed modules
- Same staff frontend; module filter by host + plan

Phase T2 — Staff auth/tenant

- Central login on each staff host
- JWT contains company + entitlements
- Enforce: host product must be entitled

Phase T3 — E-com tenant for anonymous users

- Implement path slug as default (/acme/...)
- Add custom domain mapping as white-label
- Admin “copy storefront link” generates path or custom domain URL

Phase T4 — Cross-module admin on POS host

- Keep product / preorder / dashboard / sales / online-orders on POS host
- Ecom frontend remains customer-only

Phase T5 — Superset packaging

- egf host enables POS + Landscaping module pack
- Billing/plan marks company as superset entitlement
- Ecom remains separate host/frontend

8. Risks Under Third Discussion

Risk	Impact	Mitigation
Ecom host has no company	Wrong / empty catalog	Require path or custom domain
Session tenant sticky across stores	Cross-company leak	Bind tenant per tab + validate every API
User opens landscaping host without entitlement	Module leak	Host + plan entitlement gate
Superset vs single-product confusion	Support issues	Clear product packaging docs

9. Third Discussion Summary

- **Product host** decides which product UI (POS / Landscaping / EGF / Ecom)
- **Login / session** decides which company for staff apps
- **Path or custom domain** decides which company for anonymous e-com
- **Two frontends, three products**, optional **superset**
- POS host remains the admin control plane for retail + e-com operations
- Each company still isolated by its own database

End of document — Super Architecture Data, Third Discussion Format. Planning baseline for shared product hosts + post-login tenant model.