

WHAT THE CUSTOMER NEEDS



Plant 1

Part A · 500

Part B · 100



FULFILLED?



Plant 2

Part A · 650

Part C · 200



FULFILLED?



Plant 3

Part B · 150

Part C · 300



FULFILLED?

WHAT THE SYSTEM ANSWERS



Part A — 1,500

Available



No requirement match



No plant-level coverage



No answer

Duygu Unlu

Senior SAP Solution Architect

Virtual Stock, Real Constraints

Production-grade visibility between two incompatible ERPs — without breaking either.

HOW WE MADE THIS WORK



The design decisions

Why the architecture had to look the way it does



The tradeoffs

What we chose — and what we deliberately gave up



The edge cases

The scenarios that turned out to be the real design



Why we shipped it

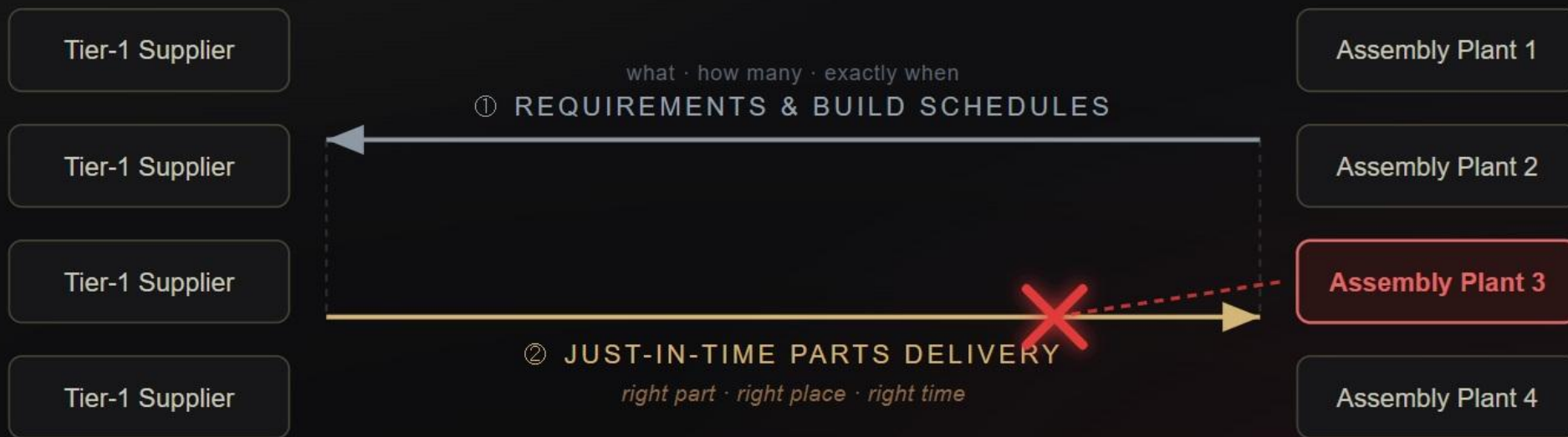
What made us confident to put it in production

● THE JUST-IN-TIME DEPENDENCY

A missing part doesn't slow the line. **It stops it.**

SUPPLIERS

OEM ASSEMBLY PLANTS



⚠ One missing part → Plant 3 line stops

\$600K – \$3M per hour

industry estimates · fully-loaded line-down cost · Siemens 2024

● REFRAMING THE PROBLEM

From a shared pool to named, committed inventory.

SUPPLIER MODEL

how the supplier sees inventory

Pool inventory

No customer assignment

"1,500 units of Part A — available"

1,500 units · undivided

Customer A

Customer B

Customer C

allocated at time of shipment



Not a transport problem.
A data modelling problem.

CUSTOMER MODEL

the OEM running Connected Supply Chain

Allocation per customer plant

"300 units of Part A — Plant 3"

Plant 1
320

Plant 2
280

Plant 3
300

Plant 4
350

Plant 5
250

1,500 units · fully divided

Three cases, no gaps. Two are automatic — **one isn't.**

A Supply = Demand ✓
Everyone covered

B Supply > Demand ✓
Covered + surplus shared by weight

C Supply < Demand
The algorithm decides who — below

Stock: **1,000** · Required: **1,550** · sorted by date, oldest first · fill until exhausted

#	PLANT	DATE	REQ.	ALLOC.
1	Plant 1	Day 1	200	200
2	Plant 2	Day 1	150	150
3	Plant 3	Day 2	300	300
4	Plant 4	Day 2	100	100
5	Plant 5	Day 3	180	180
— 70 left · next-in-queue both fall on Day 3 → tie —				
6	Plant 1	Day 3	?	contest
7	Plant 3	Day 3	?	contest

70
UNITS REMAINING
two Day-3 rows
tied for it →

TIEBREAKER CASCADE — RESOLVED IN ORDER

- 1

Earliest delivery date
Oldest requirement served first
- 2

Completeness
Same date — satisfy the one you can fully complete

FORM A
- 3

Plant priority
Both completable? Pre-agreed ranking decides

FORM B
- Rung 1 already tied (same date) → resolve by rung 2, then 3.

Rows 6 & 7 tie on date. **How they break depends on their size** — two possible forms of the same contest:

2 Completeness decides
130 can't finish in 70, so it gets zero — not a useless partial.

Form A · requests differ — only one fits in 70

Plant 1 · Day 3	needs 130 → 0
Plant 3 · Day 3	needs 70 → 70

3 OEM plant priority decides
Completeness can't separate them, so pre-agreed rank does.

Form B · requests identical — both fit in 70

Plant 1 · Day 3	PRIORITY 4	needs 70 → 0
Plant 3 · Day 3	PRIORITY 1	needs 70 → 70

Built to run in **production** — not just in theory.



Fresh data

Integration runs every 30 minutes — a live allocation view for planning and early shortage visibility.

EVERY 30 MIN



Adaptable rules

Business logic is parametric, not hardcoded. Priorities shift without major redevelopment.

PARAMETRIC



Two-sided visibility

Both sides see the same committed stock — the customer to plan, the supplier to protect it from being reassigned.

BOTH SIDES



Exception fallback

A manual trigger pushes an updated view to the customer instantly when it's needed.

ON DEMAND

When two systems model reality differently, integration isn't about moving data

—

it's about creating a *version of the truth both sides can trust.*

Thank you.

Questions welcome · duygu.unlu@fametechinc.com