

# TUM AI: Hackathon Challenge

## "C-Materials Ordering for the construction site"

### Context

comstruct is a data platform for material procurement in construction, connecting construction companies with their material suppliers and digitizing processes like ordering, delivery notes and invoices.

So far, many workflows focus on **A-materials** (high-value, project-critical items like concrete, steel, windows, HVAC equipment). Now we want to crack the messy, overlooked part of procurement: **C-materials**.

### What are C-materials?

In procurement, spend is often divided into A / B / C categories:

- **A-materials:** high value, low variety, relatively few suppliers.
- **C-materials:** low value per line, huge variety, many suppliers, tons of small orders.

C-materials are also called **tail spend**: typically only ~5% of total purchasing volume, but around **60% of all orders, 75% of suppliers and 85% of items**. ([manutan.com](https://www.manutan.com))

In construction, C-materials include things like:

- Consumables: screws, plugs, nails, tapes, foils, sealants, spray cans
- Site supplies: PPE, gloves, masks, batteries, drill bits, small tools
- "Stuff the foreman buys at the merchant on the way to site"

Individually cheap, but operationally expensive because every phone call, email, Excel and manual invoice check costs time and creates errors.

## Your mission:

Design and prototype a solution on top of comstruct that makes ordering C-materials **ultra simple for the foreman** and **fully controllable for procurement**, while tapping into supplier catalogs via contracts, Excel and APIs (PunchOut / IDS).

---

# The Challenge

## 1. Introduce an ordering flow for C-materials

Build an end-to-end ordering experience for C-materials that:

- Works for **non-digital natives** on the jobsite (foreworkers / foremen).
- Handles the fact that they **often don't know exact product names or SKUs**.
- Fits into comstruct's role as the **data / procurement backbone** between construction companies and suppliers.

Your solution should cover at least:

- **Product discovery:** How does a foreman find "the right screw for X" without knowing the SKU?
- **Cart & order creation:** How do they build and submit an order in < 1-2 minutes?
- **Feedback & status:** How do they see what's approved, ordered, delivered?

## 2. Explain C-materials clearly in the product

Foremen and even some project managers might not know the A/B/C terminology.

Design how the system:

- Explains, in simple language, **what C-materials are** and why they're ordered here (vs A-materials).
- Makes it obvious which materials go into this flow (consumables, site supplies, etc.).
- Prevents misuse (e.g. people trying to order major A-materials via the C-materials flow).

This can be via microcopy, onboarding screens, tooltips, or the information architecture itself.

---

## What You Have to Think About

### 1. Getting products & prices into the application

C-materials come from many suppliers, with different data quality. Comstruct already specializes in integrating suppliers via EDI, email scraping and portals.

For the hackathon, you can assume we ingest data through two main channels:

#### (a) Extraction from contracts / Excel

- Framework contracts, price lists and Excel files from suppliers.
- You'll need to think about:
  - Mapping supplier SKUs to a **normalized product model** (categories, units, packaging).
  - Handling **different price structures** (contract prices, discounts, project-specific prices).
  - Keeping catalog data **small and relevant** for C-materials (no BIM model of a door, just a screw).

#### Ideas to explore:

- Simple **upload + mapping assistant**: upload Excel, map columns to fields (SKU, description, price, unit, product group).
- Auto-categorization using simple rules/ML (e.g. "if description contains 'Torx' + '4,5x40' → wood screw > fastening").
- A minimal **catalog admin UI** where procurement can clean up, rename and group C-materials.

[sample.csv](#)

[fake\\_contract\\_products\\_with\\_logo.pdf](#)

## (b) API integration (PunchOut / IDS)

PunchOut catalogs let buyers “jump out” from their procurement system to a supplier’s webshop, build a basket, then send that basket back to the system as an order.

In the German-speaking construction world, **IDS Connect (Integrierte Datenschnittstelle)** is widely used to link trade software with wholesalers’ shops and exchange carts, individual prices and real-time availability.

For the hackathon, you don’t have to implement full standards, but:

- Design how **PunchOut / IDS-style flows** would look in comstruct:
  - Foreman starts in comstruct → jumps to supplier catalog → fills cart → returns cart.
  - Prices and availability are **fetched in real time**, not maintained manually.
- Consider **fallbacks** when suppliers don’t support APIs yet (e.g. cached catalog, email orders).

### Ideas to explore:

- A **“Connect supplier” wizard**: choose supplier → select integration type: Excel upload / PunchOut / IDS.
- Clear separation for the user: “Browse internal catalog” vs “Open supplier shop (PunchOut/IDS)”.
- UX for **importing a basket** from a supplier shop and adding project metadata (project ID, cost code, delivery location).

For testing, we created this simple webshop where you can add a return url to jump back to your application with the cart.

## 2. Handling many products & suppliers

C-materials have huge assortments. The challenge: **don't dump 50,000 SKUs on the foreman.**

**Ideas to explore:**

- **Product groups & kits:**
  - "Drywall kit for 50 m<sup>2</sup> wall" → predefined bundle of screws, anchors, tape.
  - "Basic PPE pack for new worker" → helmet, gloves, glasses.
- **Use-case based search** instead of SKU search:
  - Search by "task" or "building element" ("fix gypsum board to metal stud", "seal window") and get recommended items.
- **Favorites / recent orders:**
  - Foreman sees: "Most ordered on this project", "Your last 10 items", "Company favorites".
- **Smart filters:**
  - Filter by trade, product group, supplier, project standard.
- Optional: "Good / better / best" variants (low/medium/premium) to limit choice but keep some flexibility & use AI!

Research on construction SaaS UX emphasizes focusing on concrete on-site workflows, fast task completion and minimizing data entry, especially for foremen.

---

## 3. Easy ordering for the foreworker / foreman

Target persona:

- Not a digital native.
- On site, often with gloves, dust, poor reception.
- Time-poor: if it takes longer than a phone call, they won't use it.

**Design goals:**

- **Mobile-first, 2–3 taps to re-order** standard C-materials.
- Works on **small screens**, with large tap targets and minimal typing.
- Can be used in **short bursts** (between tasks, in the container, on the scaffold).

#### Ideas to explore:

- **"Shopping list" style UI:**
  - Categories with big icons (Fasteners, Consumables, PPE, Tools...).
  - For each item: + / – buttons to set quantity, no modal dialogs.
- **Order from templates:**
  - Project manager defines "C-material sets" per project phase; foreman just chooses set + tweaks quantities.
- **Guided order wizard:**
  1. What trade / task? (e.g. drywall, tiling, concrete work)
  2. Approx. quantity / area?
  3. Suggest default C-materials list, with adjustable quantities.
- **Helpful copy instead of technical language:**
  - "Small everyday items & consumables for your site" instead of "Class C materials".

---

## 4. Easy controlling for procurement

Procurement wants control without blocking the site.

### 4.1 Restricting orderable products

Your solution should allow procurement to:

- Define **which product groups** or suppliers are orderable as C-materials.
- Maintain **preferred suppliers** and hide non-preferred ones.
- Configure **company / project-specific assortments** (only approved items appear for that project / trade).

This aligns with how IDS and similar interfaces let companies expose specific assortments, prices and availabilities to their trade partners.

**Ideas to explore:**

- A **procurement console** where:
  - They can toggle product groups on/off per project.
  - Mark items as "Allowed for C-materials" / "A-material – order via main workflow".
  - Set default suppliers per category (e.g. all PPE from Supplier X).

## 4.2 Approval & budget control

For **bigger orders**, procurement wants to check before anything is committed.

**Ideas to explore:**

- **Approval thresholds:**
  - E.g. orders < 200 CHF → auto-approve.
  - Orders ≥ 200 CHF or certain product groups → send to approver.
- Different flows for:
  - **Project manager approval** (for project budgets).
  - **Central procurement approval** (for framework compliance).
- Clear visibility for the foreman:
  - "Draft → Pending approval → Approved → Ordered → Delivered."
- Basic **spend analytics:**
  - C-material spend per project, per supplier, per product group.
  - Which projects / foremen generate the most tail spend?

Comstruct already focuses on giving real-time data for controlling, automated invoice verification and ESG reporting.

---

## What We Expect as Hackathon Output

You don't need a full production system – focus on a convincing concept and a working slice.

**Minimum:**

**1. User journey & UX prototype**

- Mobile-first flow for a foreman ordering C-materials.
- Procurement view showing how they configure assortments and approvals.
- Clear explanation in the UI of what C-materials are.

**2. Technical concept**

- How product & price data flows in:
  - From contracts/Excel.
  - Via PunchOut / IDS-style API connection.
- How you would model C-materials (product groups, attributes, pricing).
- How approval rules and restrictions are stored & enforced.

**3. Demo scenario**

- Example project.
- 1–2 example suppliers (one via "Excel upload", one via "API/PunchOut/IDS").
- End-to-end demo: foreman places C-materials order → procurement reviews / approves → order confirmed.

**Nice-to-have:**

- Simple mapping/extraction prototype (e.g. upload Excel and see normalized products).
- "Use-case search" (task-based) instead of pure text/SKU search.
- Offline-friendly design / graceful handling of poor connectivity.

---

## Judging Criteria



- **User fit (Foreman)** – Is it obviously easier than a phone call/WhatsApp? Could a non-digital native use it without training?
  - **User fit (Procurement)** – Do they get real control (assortments, budgets, approvals) without micro-managing each order?
  - **Handling of complexity** – Smart ways to deal with huge assortments, many suppliers and messy data.
  - **Integration thinking** – Realistic concepts for Excel/contract ingestion and PunchOut / IDS-style integrations.
  - **Clarity of the C-materials concept** – Is it crystal clear which materials belong here and why?
-