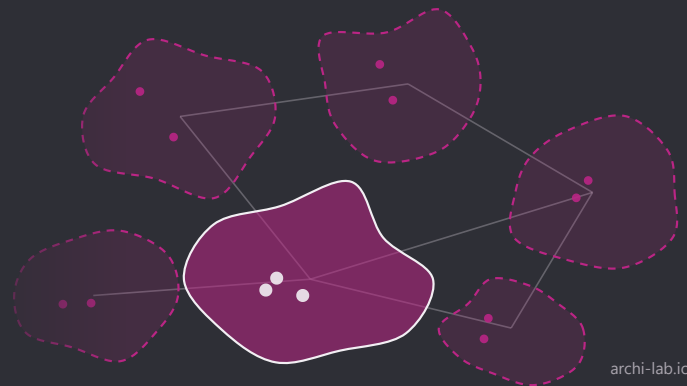


MASTER ELECTIVE · WINTER SEMESTER 2026/27

DDD

Domain-Driven Design of Large Software Systems



Kickoff

Friday, 02.10.2026

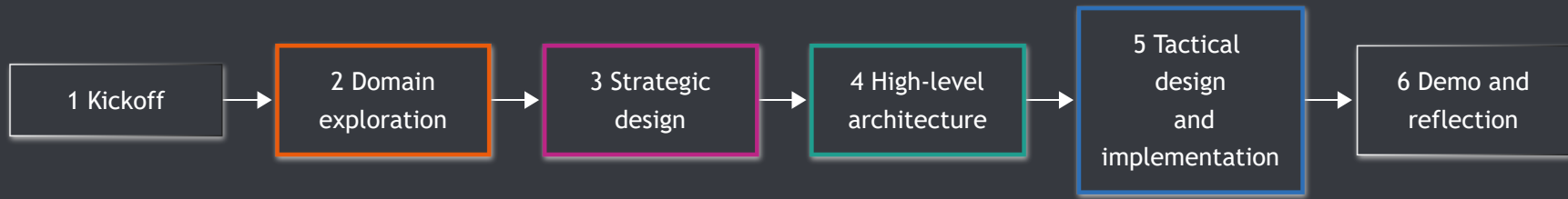
What this course is about

You design a reasonably complex greenfield system for several teams, using Domain-Driven Design, and you build a first version of it.

- Explore a business domain together, with the people who know it
- Cut it into bounded contexts that teams can build independently
- Turn each context into a service in a given microservice architecture
- Model and implement your own context, with an AI agent kept on track by guardrails you write
- Hand over an architecture that someone who was not in the room can work with

- The architecture you hand over is a deliberate first cut
- Reviewing and evolving it is the job of the *AI-based Architecture Engineering* (ARCEN) module
 - (expected to start summer semester 2028)

Six phases



Design, October to early November

EventStorming, strategic design with domain stories and the context map, and a C4 model of a microservice architecture. Each of these days is moderated by one of you: a specialist team.

Build, November to February

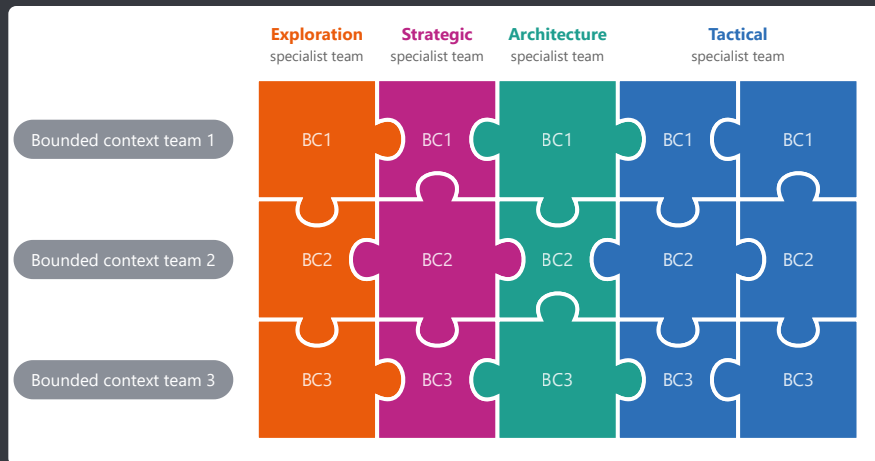
Your team implements its bounded context as a service, first on its own, then improved with what the two tactical workshops teach. The semester ends with a demo and a reflection on the method.

You are in two teams at once

We follow the jigsaw approach to cooperative learning.

- A **specialist team** owns one phase: it studies its method, moderates that phase's workshop, and coaches everyone on it
- A **bounded context team** owns one part of the case study, from specification to service
- Every bounded context team holds one member of every specialist team

Whoever prepared the context map workshop sits in your team when you draw your own boundary.



The four specialist teams

Exploration Team

Make the domain visible to everyone. Moderates the EventStorming day on 16.10. Afterwards: guardians of the discovered events: they speak up when an event from the board goes missing.

Architecture Team

Turn the contexts into a running landscape. Documents the provided stack, moderates the C4 day on 06.11., takes the cross-service technology decisions with the supervisors. Afterwards: keepers of the shared infrastructure and the architecture decision records.

Strategic Team

Cut the domain into independent contexts. Moderates strategic design on 23.10. Afterwards: owners of the context map and the domain stories, including the demo scenarios.

Tactical Team

The largest team, with two focus areas: guardrails for the DDD building blocks, and the agentic AI harness for team development. Starter platform out on 06.11., workshops on 27.11. and 04.12. Afterwards: platform owners and aggregate coaches. Staffed from those who already know the DDD building blocks.

Each team has a full charter on the course page: mission, first steps, deliverables, standing role, and support by the supervisors.

Exploration Team

You make the domain visible to everyone, and keep its language honest.

YOUR PATH

From today

Research EventStorming. Plan a big-picture session for the whole class, with two parallel sub-boards and a merge if we are many. Prepare the questions that bring misunderstandings to the surface.

16.10.

You moderate the EventStorming day: the whole class meets the domain for the first time, and you run the room.

Afterwards

Guardians of the discovered events. When an event from the board goes missing in a domain story or an event API, you speak up.

WHAT YOU TAKE AWAY

- Facilitating a room of people who disagree about a domain: the everyday skill of whoever runs requirements or architecture workshops in a real project
- EventStorming, a method teams use to start new systems and to untangle legacy ones
- An ear for language: where one word means two things, a boundary is often hiding

Good fit if you enjoy working with people as much as with code, and can keep a room moving.

Exploration Team: what 16.10. looks like



Timelapse: Mariusz Gil, github.com/mariuszgil/awesome-eventstorming

Strategic Team

You cut the domain into contexts, the decision every service has to live with.

YOUR PATH

From today

Research the core domain chart, domain storytelling and the context map.
Observe the EventStorming day and note where the boundaries might run.

23.10.

You moderate strategic design. The class draws the contexts and picks two or three domain stories as the scenarios the system has to run at the demo.

Afterwards

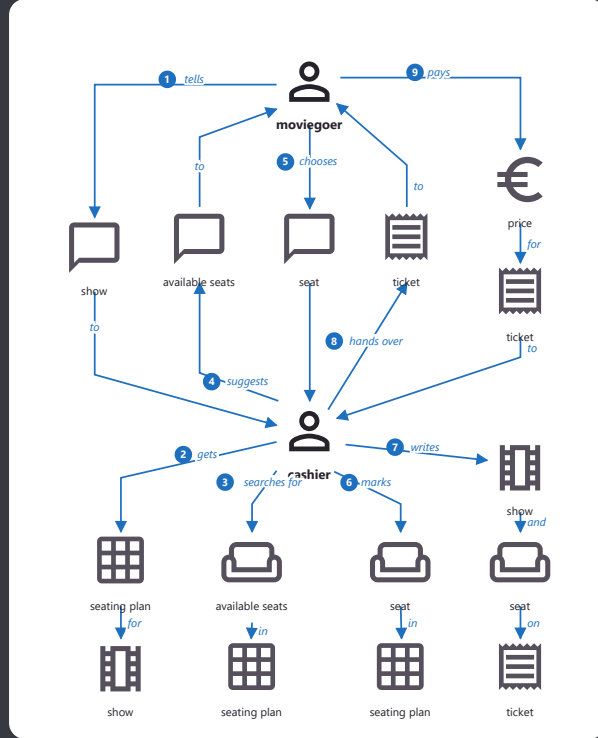
Owners of the context map and the domain stories. When teams change what flows between their contexts, you keep the cut coherent.

WHAT YOU TAKE AWAY

- The boundaries you draw with the whole group decide what the services are, how tightly they are coupled, and how far a change spreads
- Domain storytelling and context mapping: methods for talking about a system with the people who run the business
- Keeping a model consistent while four teams build on it

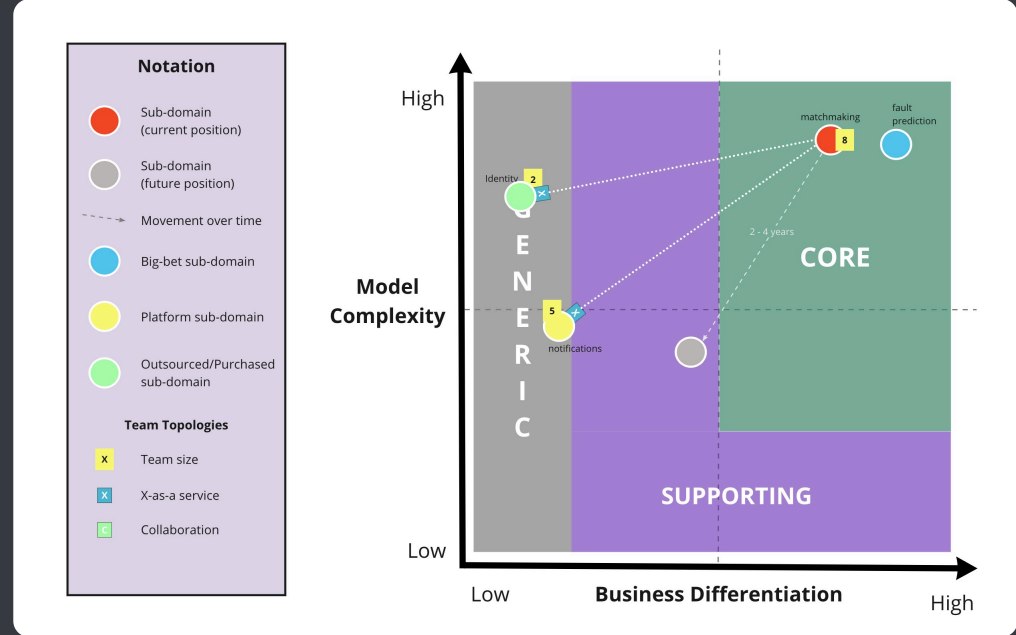
If you want to do architecture: the cut into bounded contexts is the most far-reaching architectural decision in this course, and it is made here.

Strategic Team: the tools of 23.10.



Domain Storytelling: who does what with what, in numbered steps

Domain story: "Metropolis" cinema example from S. Hofer, H. Schwentner, Domain Storytelling (2021), github.com/WPS/egon.io-examples, redrawn. Core domain chart: DDD Crew, github.com/ddd-crew/core-domain-charts, CC BY-SA 4.0.



Core domain chart: core, supporting, generic

Architecture Team

You make the shared infrastructure usable for everyone, and record the decisions.

YOUR PATH

From today

Learn the provided stack: Kafka and PostgreSQL in Docker, the helper libraries for the outbox pattern and for Kafka producers and consumers. Document it, prepare a demo, research C4 levels 1 and 2.

06.11.

You moderate the high-level architecture workshop and agree the technology decisions that go beyond a single service with the supervisors.

Afterwards

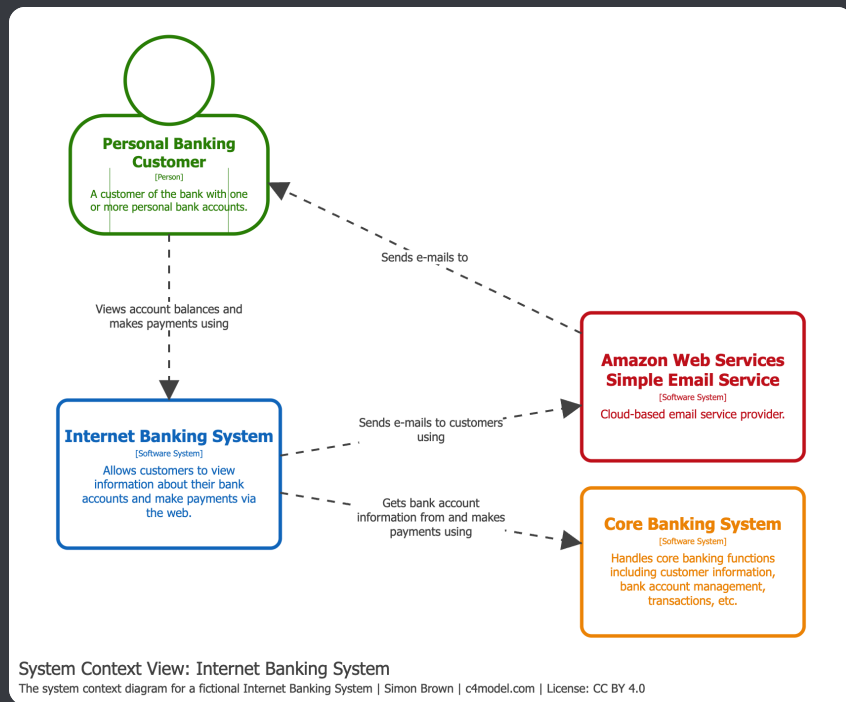
Keepers of the shared infrastructure, administered and documented together with the supervisors, and of the architecture decision records.

Much of this work is infrastructure and documentation. The other architectural decisions are made elsewhere: the context cut by the strategic team, the rules for the code by the tactical team, the integration by the bounded context teams.

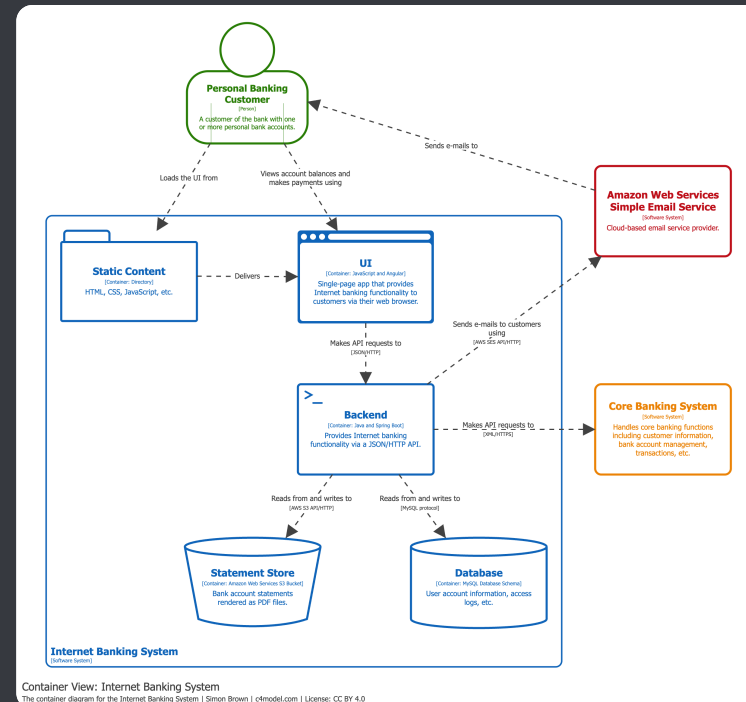
WHAT YOU TAKE AWAY

- Hands-on experience with Kafka and the outbox pattern, the plumbing of event-driven systems
- C4 diagrams and architecture decision records, the documentation practices teams actually keep
- Explaining a technical platform so that others can build on it

Architecture Team: C4, levels 1 and 2



Level 1: system context



Level 2: containers

Tactical Team 1 / 2: guardrails

The largest team, with two focus areas. The first: the rules every service's code lives by, checked on every build.

YOUR PATH

From today

Build the starter platform: repository template, build pipeline, and the first guardrails as automated checks, for layering and for context boundaries.

06.11.

You hand the platform out. No team starts from an empty repository.

27.11.

Workshop on the DDD building blocks: a supervisor teaches, you moderate the exercises and coach. You write the aggregate-level guardrails while teaching them.

Afterwards

Platform owners and aggregate coaches inside the bounded context teams.

WHAT YOU TAKE AWAY

- Architecture as code: a rule only holds if the build can tell when it is broken, and writing that check means understanding the rule precisely
- Entities, Value Objects, Aggregates, Repositories and domain events, at a depth that only teaching them gives you
- The most direct architectural influence in the course: your checks run on every build of every service

Staffed from those of you who already know the DDD building blocks: a rule has to be known well before it can be checked automatically.

Tactical Team 2 / 2: Agentic AI harness (based on Claude Code)

The second focus area: how a whole team builds software with an Agentic AI (Claude Code), not just one person.

YOUR PATH

From today

The Claude Code setup of the starter platform: `CLAUDE.md` files with the project's conventions, skills for recurring tasks (a new aggregate, a new event), hooks that run the build and the ArchUnit guardrails before a task counts as done, subagents for reviews, and permission settings for what the agent may touch.

04.12.

You moderate the second tactical workshop, on a proper setup for working with Agentic AI. The teams rework their setup with what you show them.

Afterwards

Help improving the Agentic AI harness wherever the teams find the AI going wrong.

WHAT YOU TAKE AWAY

- What an agent needs to know about a codebase (`CLAUDE.md`, skills), what it must not touch (permissions), and how its output gets checked (hooks, review subagents)
- Guardrails as the AI's feedback loop: an agent that runs the architecture checks corrects its own mistakes
- Team-scale AI-assisted development, something most companies are working out right now

Good fit if you already work with AI coding agents and want them to work for a whole team.

Tactical Team: guardrails the agent runs into

```
@ArchTest
static final ArchRule contextsDoNotShareDomainClasses =
    noClasses().that().resideInAPackage("..ordering.domain..")
        .should().dependOnClassesThat()
        .resideInAPackage("..kitchen.domain..")
        .because("bounded contexts integrate via events, "
            + "not via each other's domain classes");
```

A guardrail: an architecture rule as a test, with the reason built in.

Agent writes code



Hook runs build and guardrails



Violation, with the "because"



Agent fixes it, build is green

Claude Pro for Your Project Work

While you're working on this project, you'll receive **free access to Claude Pro**, and with it, Claude Code.

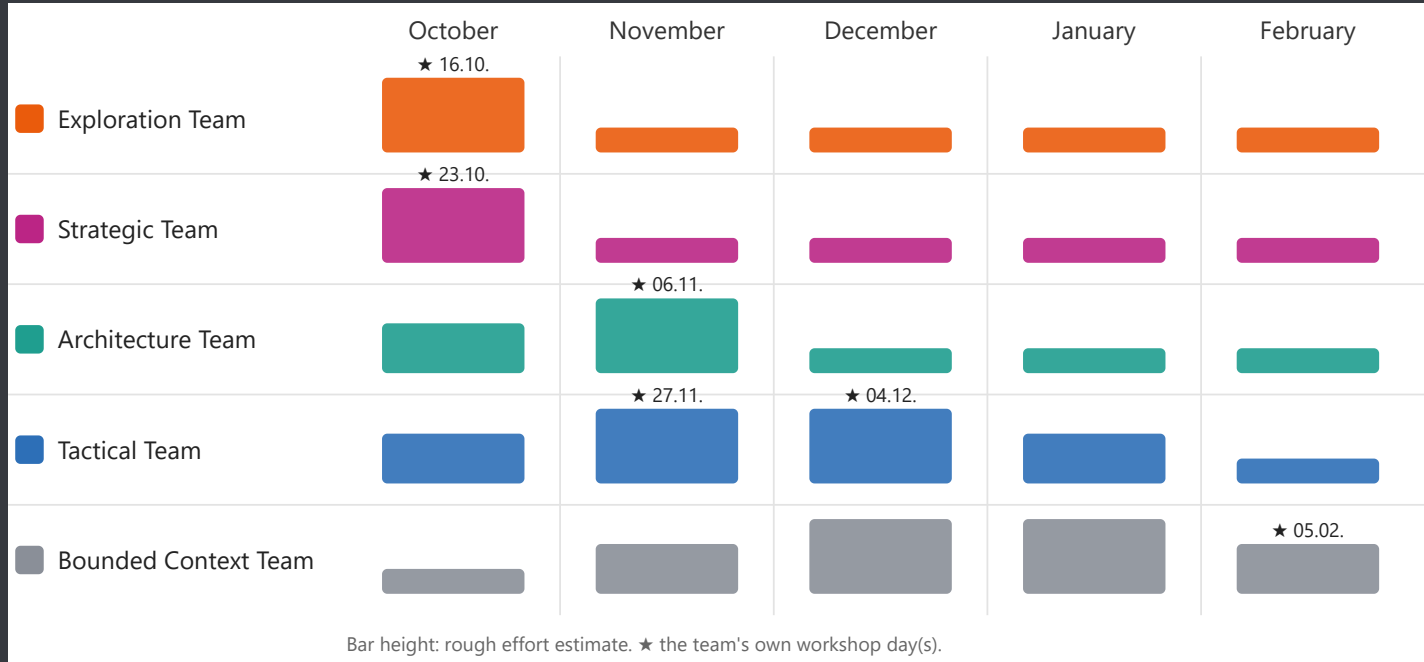
- Made possible by the Anthropic Team Plan for Scientists
 - for 10,000 scientists worldwide - I'm one of them ;-)
- How to work in a team with agent-based AI is a (sub)topic of the module
- The **Tactical Team** will be responsible for conceptualizing and coaching the other bounded context team members in how to properly use agentic AI in a team setting

How to choose your specialist team

1. Orientation: a short quiz on the course page, ["Which Specialist Team Suits You?"](#)
2. Vote, now: in the [survey](#) linked on the course page
 - YES for your first choice team(s)
 - MAYBE for teams you would accept
 - NO for teams you'd rather not be in
3. Today: we evaluate the survey on the spot, and all teams are formed before we leave the room.

The tactical team should ideally be staffed from those of you who already know the DDD building blocks.

When each team is busy



Your own load is your specialist team's row plus the grey row. All specialist teams start right after the kickoff.

The bounded context team

Bounded context teams are responsible for one specific part of the restaurant case study. They design and implement the service(s) for that bounded context based on the input from the previous workshops.

When	What your team does
from 23.10.	Forms at the end of strategic design; collects the events that belong to its context, the start of its event API
06.11. - 27.11.	First version of your specification: aggregate sketch, the event API of your service in AsyncAPI, and a walking skeleton that publishes one event over Kafka
from 27.11.	Rework model and setup with what the tactical workshops teach; implement the service and a demo client; keep your event API up to date
18.12. and 15.01.	Show at the status meetings how far the demo scenarios run end to end: integrating is your team's job
05.02.	Demo, reflection, handover package

Dates

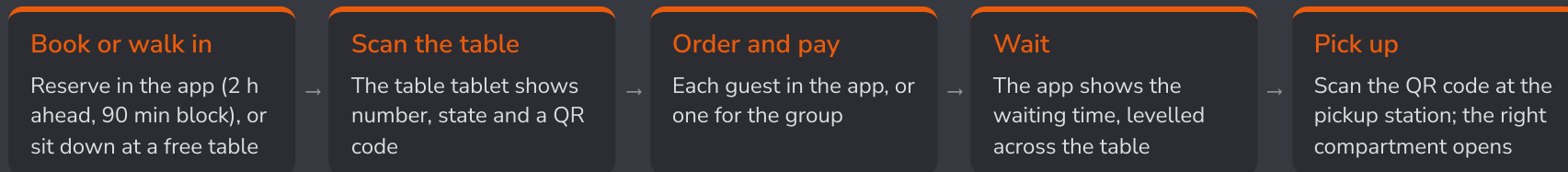
Fridays, 10:00 - 17:00, on site. The two status meetings are half days, on site or via Zoom.

Date	Phase	Workshop	Moderated by
02.10.2026	1	Kickoff	Supervisor(s)
16.10.2026	2	EventStorming "Big Picture"	Exploration Team
23.10.2026	3	Strategic design: domain stories and context map	Strategic Team
06.11.2026	4	C4 model of the microservice architecture	Architecture Team
27.11.2026	5	Tactical design: the DDD building blocks	Tactical Team
04.12.2026	5	Tactical design: a proper setup for working with an AI agent	Tactical Team
18.12.2026	5	Status meeting (1), half day	Supervisor(s)
15.01.2027	5	Status meeting (2), half day	Supervisor(s)
05.02.2027	6	Demo and retrospective	Supervisor(s)

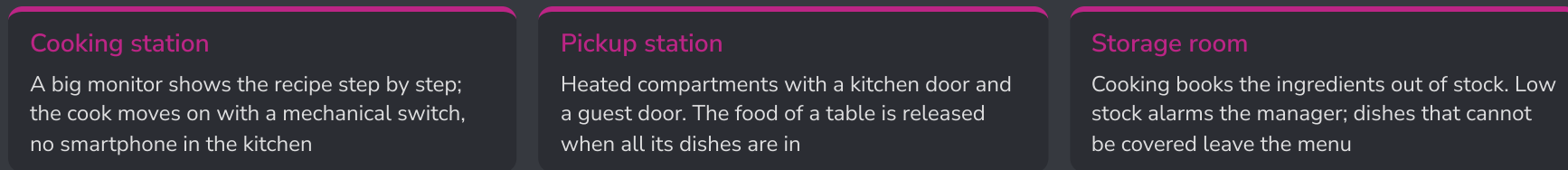
The case study

Delicious, a franchise chain of restaurants without waiters and without a counter. One system for all its restaurants.

THE GUEST'S WAY



BEHIND THE SCENES



The manager runs menu, recipes, cooks and stock in the Admin Client. Drinks (an outsourced bar), the duty rota and ordering supplies are out of scope. Here is what that customer told you in a kickoff meeting: [archi-lab.io · Case Study](https://archi-lab.io/case-study).

The technical setting

- The architectural style is *given*, not chosen
 - *Microservices* integrating over Apache Kafka
 - one or several service(s) per bounded context
- The supervisors provide the main infrastructure
 - Kafka
 - PostgreSQL in Docker
 - Helper libraries for the outbox pattern and for Kafka producers and consumers (Java)
- The tactical team provides the starter platform
 - repository template, pipeline, agent configuration, architectural guardrails
- A service is done when its tests are green, its guardrail pipeline is green, and it has no open merge requests

Your effort should go into the domain, not into plumbing.

Build first, improve later

- After 06.11. your team starts building right away, on the starter platform
- The two tactical workshops come after that, as corrective steps
 - The DDD building blocks on 27.11.
 - A proper setup for working with an AI agent on 04.12.
- You will rework parts of what you built.
 - That is intended: it is how most real teams encounter
 - DDD is about working in iterations
- Reworking in response to what you learned is *rewarded* in the grading, not penalised

Grading

Part	Criterion	Weight
Specialist role (per specialist team)	Preparation and moderation of your phase	2
Specification (per context team)	The DDD artefacts for your context	4
Implementation (per context team)	Platform readiness	precondition
	Your context as a service	4
	Working method with agent and guardrails	2
	Reviewability of the handover	2
Reflection (per context team)	Reflection on the method	1
Participation (individual)	Contribution beyond your own team	1
	Sum	16

- Each criterion scored 0 to 3, so 48 points in total
- A third of the maximum passes (4,0), eleven twelfths give a 1,0
- Team points can be reduced for a member who contributes clearly less, and nobody gets more than 100 % of the team result
- Passing requires your service to build on the platform with a green guardrail pipeline

One request: research consent

- I would like to study how this course design works, and possibly publish about it
- For that I need your permission, now at the start, to ...
 - Analyse your course artifacts anonymously after the course
 - Evaluate your reflection at the end of the course
 - Possibly conducting (anonymous) surveys
- A short, named survey in ILU:
 - Voluntary
 - Without any influence on your grade
- Please go to this [ILU survey](https://ilu.th-koeln.de/goto.php/svy/902873) and give (or deny) your consent for participation in that study



Now, on your phone or laptop
ilu.th-koeln.de/goto.php/svy/902873

Next Steps (after Today's Workshop)

- Everyone individually:
 - Join the ILU course for DDD, if you have not done so already
 - Register for taking the DDD exam in PSSO (registration is between 5.10. and 26.10.)
 - but only if you are actually admitted to the course (in case we have to prioritize)
 - Make yourself familiar with the [ArchiLab website](#) and the DDD material you find there
- Exploration team:
 - Since your workshop is coming up soon, start reading the event storming material
- Every expert team:
 - Use my [Calendly booking page](#) to book a planning status meeting
 - Latest in the weeks starting the 12th or the 19th of October

After lunch: how the teams are formed

Your survey votes go into an algorithm that we run here in the room. It computes the best assignment exactly, and anyone with the votes and the seed can reproduce it.

- 1 **Team sizes** The teams differ by at most one member; the Tactical Team is twice the size of the others.
- 2 **Yes first** As many of you as possible get a team you voted Yes for. Each Yes counts once, however many teams you said Yes to.
- 3 **Your Nos are one veto in total** One No is a full veto; No on three teams is a third of a veto each. Rejecting everything but one team buys no priority.
- 4 **Then a public lottery** For the remaining ties. One of you calls out the seed; the result shows where the lottery decided.
- 5 **Then Maybe before No** If a team has to be filled, a Maybe goes there before a No.

How to vote: Yes where you would really like to be, Maybe generously, No only where you mean it.

Your preferences, now

- Not voted yet? Vote now.
- Changed your mind after this morning? Update your vote.
- Please be done before the lunch break ends: after lunch, we form the teams.

Yes / Maybe / No for each of the four specialist teams.



On your phone or laptop
xoyondo.com/op/bs153qy1cv9tmzr