

Building Bridges:

From
Structural Engineering
to
Strategic Design

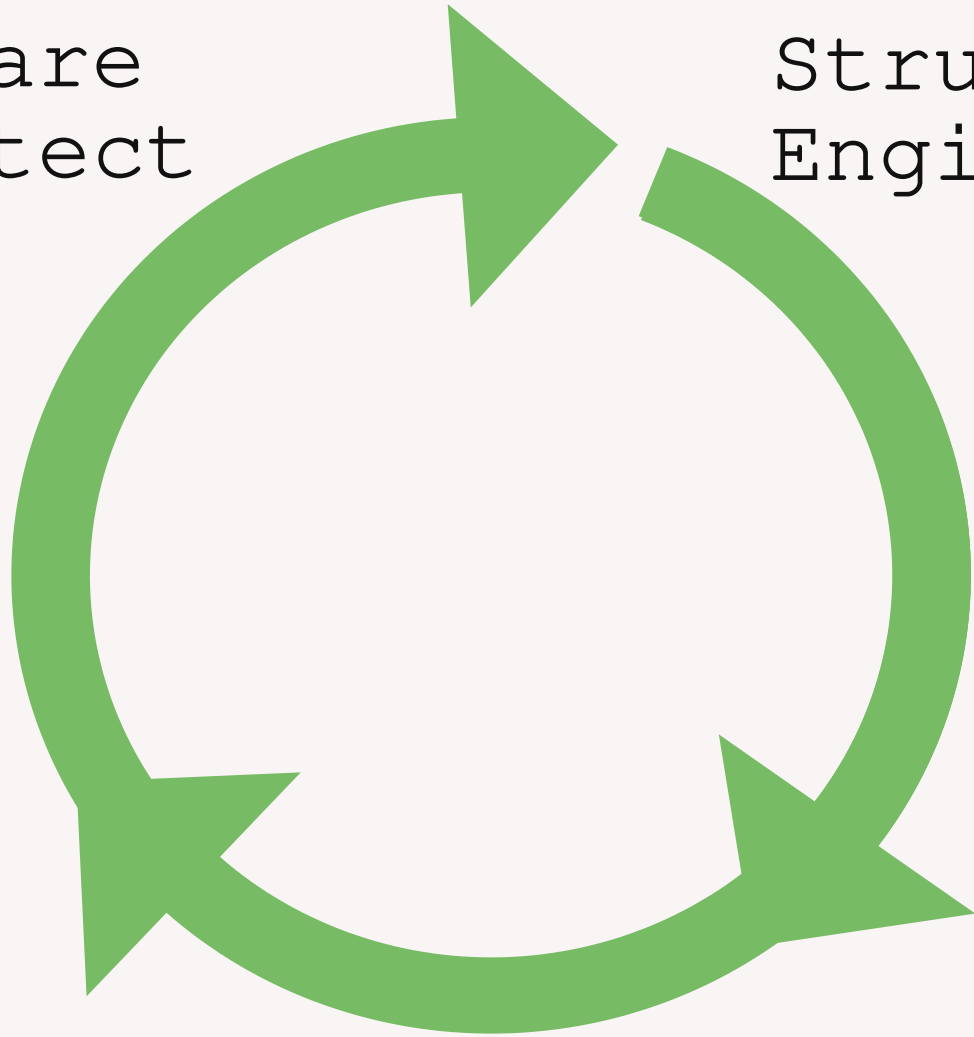
Katharina Damschen

Software Architect & Consultant

Software
Architect

Structural
Engineer

factor 10



Software
Engineer

Structural
Engineering
Application
Developer

There are so many
bridges in Software
Architecture

factor 10

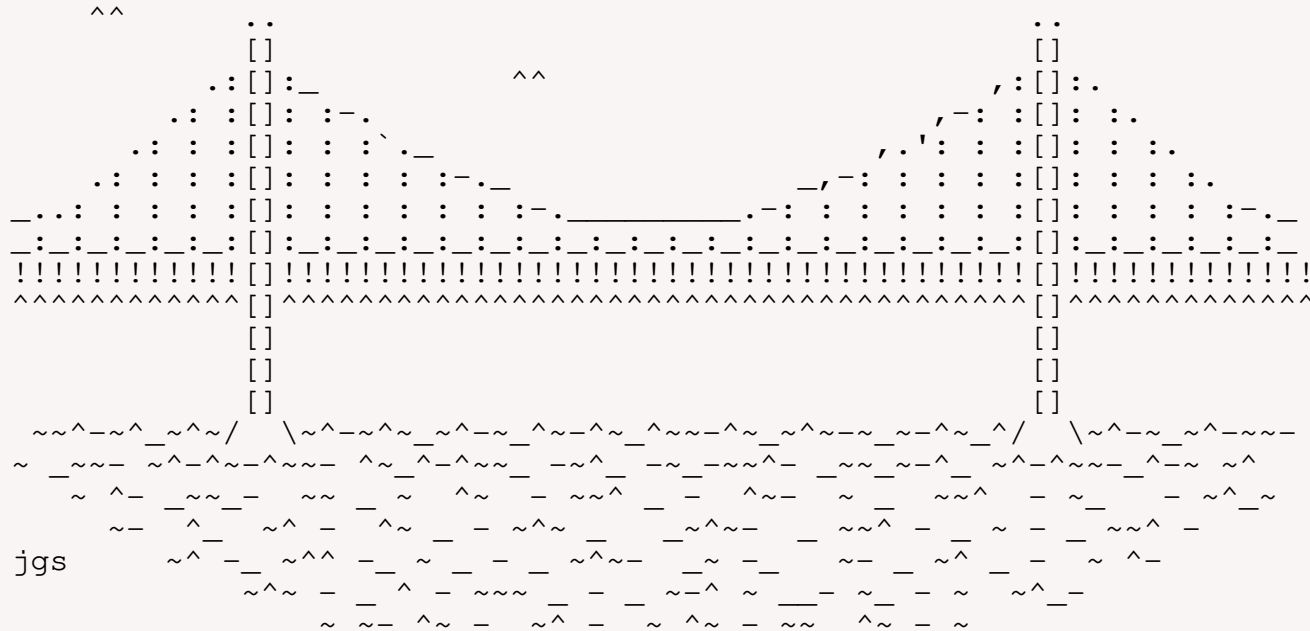
.. but only few architects
in bridge design

Structural Engineering^{factor 10}

.. is in its core the study
of how forces travel through
a system and how that system
reacts to them.

The Software Structural Engineer

factor 10



Agenda

- Modeling to reduce cognitive load
- Designing for failure
- Building dependable & economic
- Simplicity & collaboration

Modeling to reduce cognitive load

factor 10

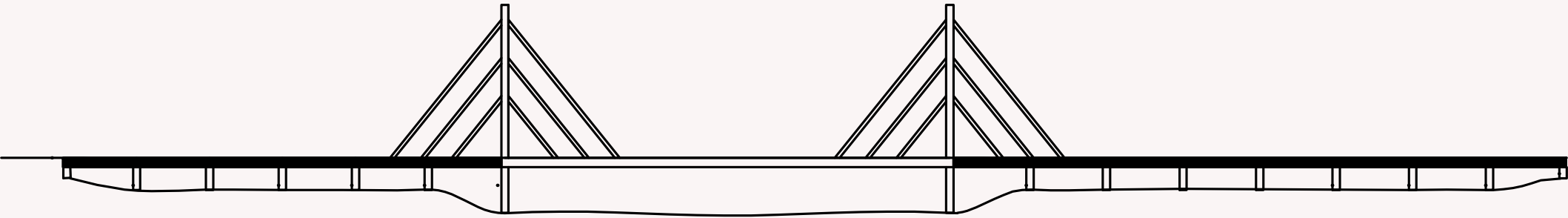


Beeckerwerther bridge, Duisburg (Germany)

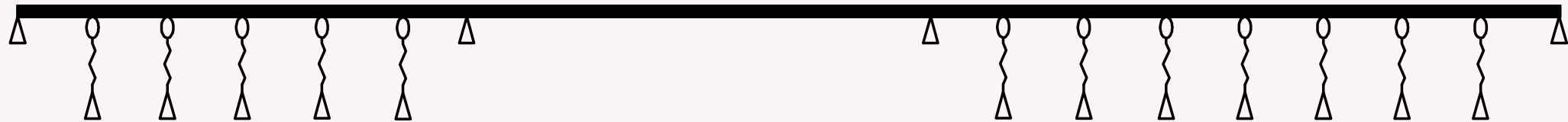


factor 10

Schematic



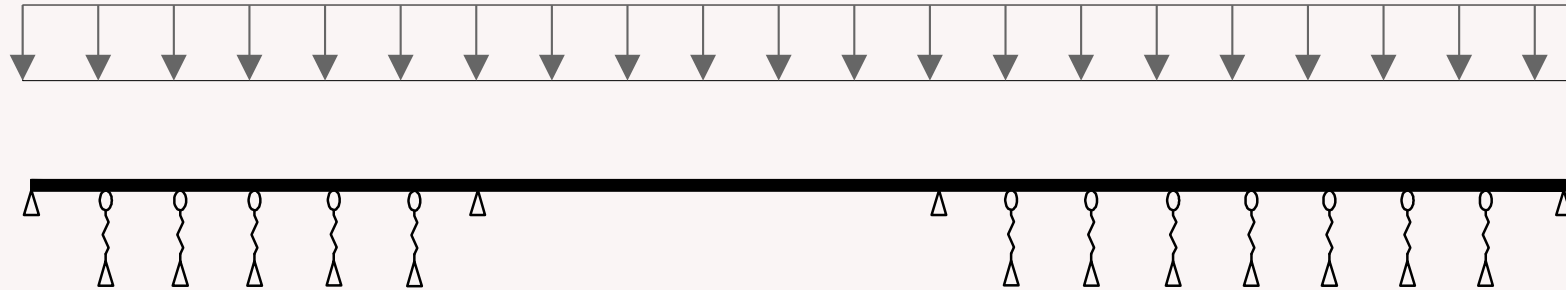
'Global Analysis'

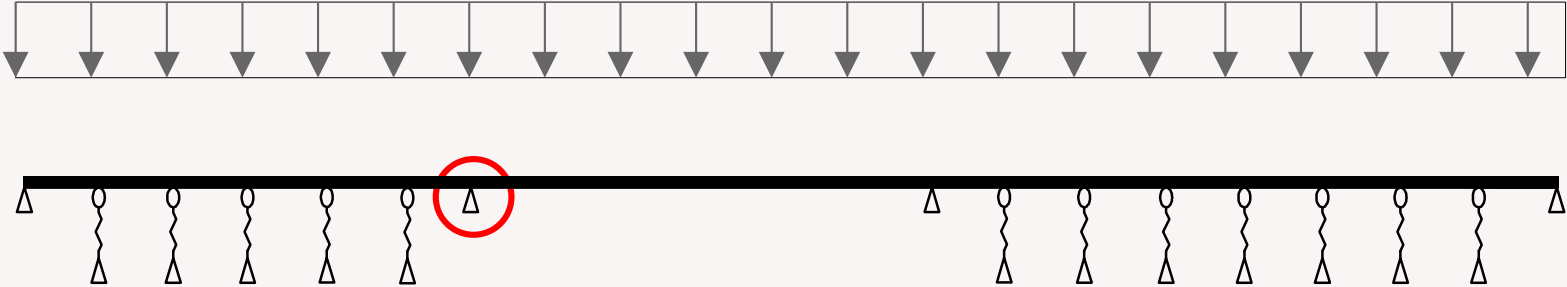


Structural Idealization

The "*Global Analysis*"

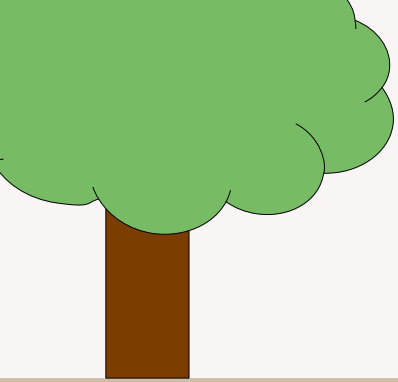
factor 10





Zooming in

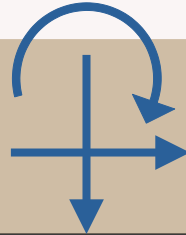
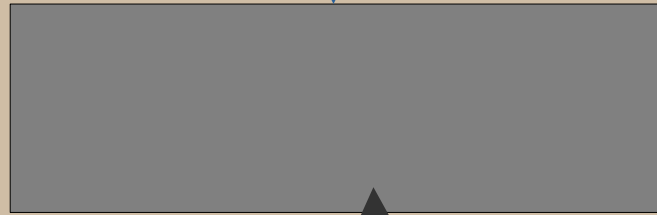
factor 10



Soil



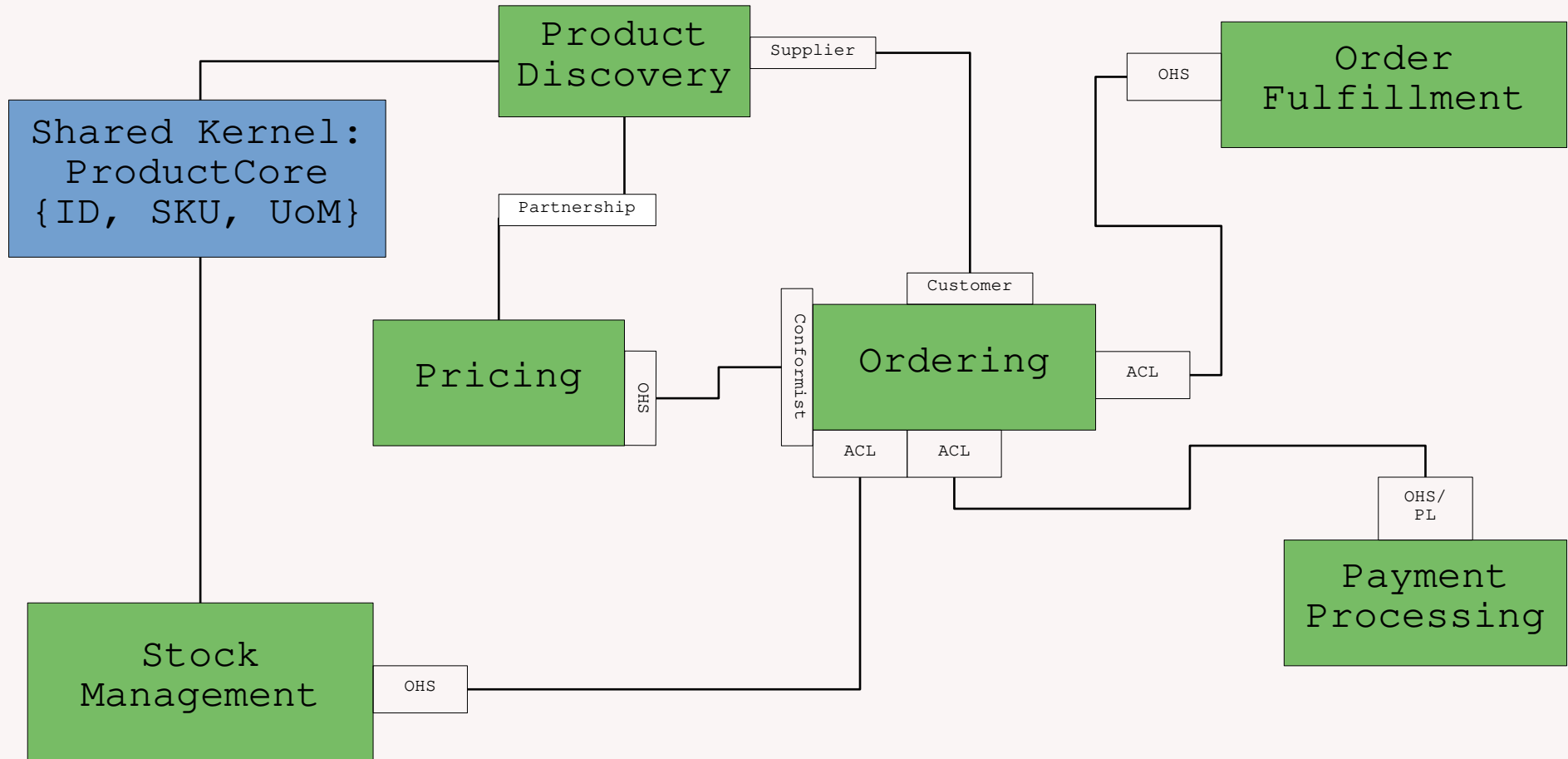
Forces applied
to foundation



Concrete foundation



Context Map: E-Commerce ^{factor 10}



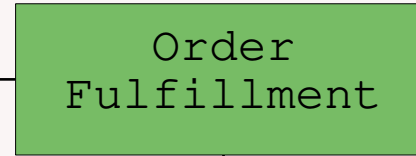
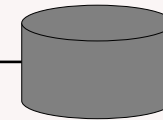
Zooming in

factor 10

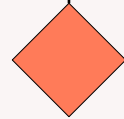
Messages:

Events
Queries
Requests

Web framework



Persistence



Hosting

Shared Principles ^{factor 10}

- Avoid God models
- Define Bounded Contexts to restrict cognitive load.
- Ensure contexts do not leak internals.
- Not all problems can be solved by looking closer at the details

Designing for Failure



Graceful Degradation

factor¹⁰





Clarify Boundaries

Reduce
Mean
time
to
recover



factor 10

Advantages that go beyond the system not failing





Building dependable **AND** economic

Seiya Maeda on Unsplash
ZEKERIYA SEN on Unsplash

Eyüpcan Timur on pexels
Dirk Erasmus on Unsplash

Design Working Life

adjust the robustness and maintainability of a system
according to its expected lifetime.

Design Situations

Persistent Design Situations

factor 10

Expected use during Design
Working life.

Factored up (safety
factors).

Building dependable & economic

Persistent Design Situations

factor 10



Accidental Design Situations

factor 10

Caused by accidents.
Do NOT factor up.

Accidental Design Situations

factor10



In which situations do you want your system **to work as intended**, and when is it okay for the system to stop working as long as the **minimum guaranteed behavior** is met?

Limit States

Ultimate Limit States

Ultimate Limit States^{factor 10}

- Geotechnical** → External Services
- Equilibrium** → High Coupling
- Strength** → Resource Constraints/Bugs
- Fatigue** → Cruft, Software Rot

Deliberate Threats

factor 10

.. introduce an additional Ultimate Limit State:

Security

Serviceability Limit States

Kritiken handlar dels om utformningen av systemet i sig, som att det krävs nio klick för att skicka i väg ett brev, dels om att utbildningen i det nya systemet är otillräcklig. Förutom det 50-tal

From: <https://lakartidningen.se/nyheter/oro-infor-millenniumstarten-pa-sas-bromsa-taget-innan-det-kraschar/> 26-04-10

The criticism concerns, on one hand, the design of the system itself, such as the fact that it takes nine clicks to send a letter, and on the other, that the training for the new system is insufficient.

already halted the planned introduction of Millennium in 2021 after hundreds of doctors voiced fierce criticism. Millennium had been developed for American and not Nordic conditions, they said.

“serious performance issues”

From: <https://www.heise.de/en/news/Scrapping-the-millennium-introduction-of-a-health-record-in-Sweden-fails-10323142.html> 2026-04- 10

Both
the Ultimate and
Serviceability limit
states **must be**
satisfied.

"Nines don't
matter when users
aren't happy."

Charity Majors

Consequence Classes

factor 10

Build in a buffer into systems
whose failure would have
extraordinary consequences.

Deliberately and consciously accept
lower robustness for lower-
consequence systems.

Recap

- Decide what to design for -> Design Working Life & Design Situations
- Decide what must not happen -> Ultimate & Serviceability Limit States
- Decide how much buffer is justified -> Consequence Classes

Zooming out

Changing the strategy from
**designing for all possible
accidents**

to

Risk Avoidance

Simplicity & Collaboration

factor¹⁰

Tiger and Turtle, Duisburg, Germany

Photo by Marvin Damschen

Make it easy to
do the right
thing.

The code is the
architecture

factor 10

Systems designed in an ivory
tower are unlikely to
survive contact with
reality.

There are Starchitects^{factor 10}



..but (Structural)
Engineering is a team effort

Empirical knowledge ^{factor 10}

Have
Postmortems!





factor 10

Design deliberately.
Fail predictably.
Learn collectively.
Iterate mercilessly.

Thank you!

factor10

katharina.damschen.net
katharina.damschen@factor10.com
linkedIn: katharina-damschen
mastodon.nu/@katharina

What can Software
Engineering teach?