blindw3rk
digitale architekten

Best Practices using Shopware for Composable Commerce Projects

About

blindw3rk
digitale architekten

Thomas

Head of Technology
–
Happily deleting legacy infrastructure to deploy high-performance, automated platforms.

Christoph

Lead Developer
–
Building high-performance e-commerce solutions that make both the CEO and the servers happy.

We are digital architects.

We plan and manage complex digital projects.

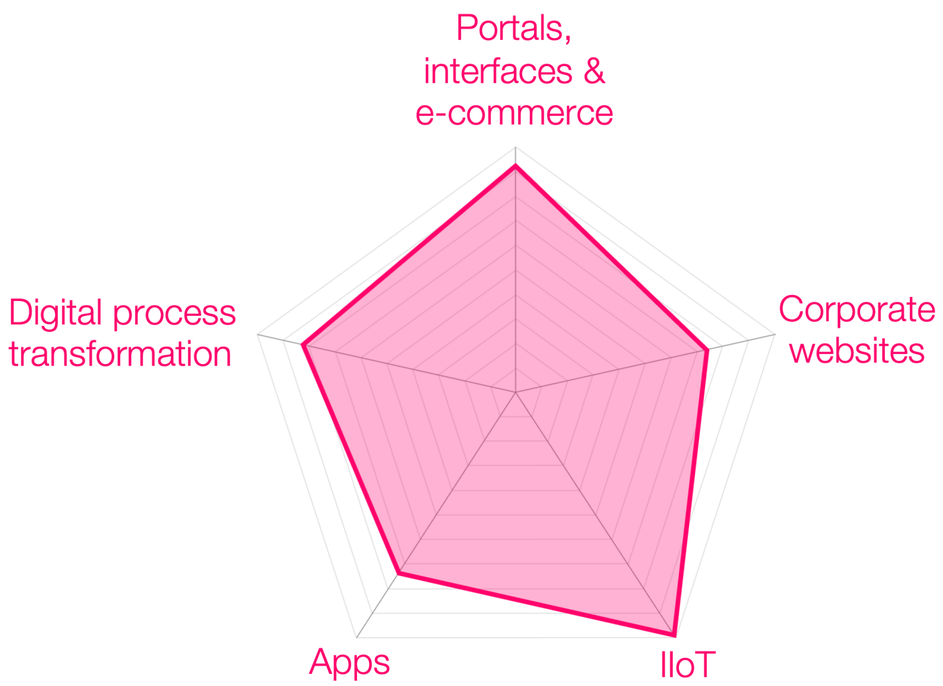
The results are digital ecosystems and scalable platforms.

blindwerk – blind What?

Staff
21 employees
65 freelancers

More than 200
national and
international clients

in the D-A-CH region



What This Talk is About

Content

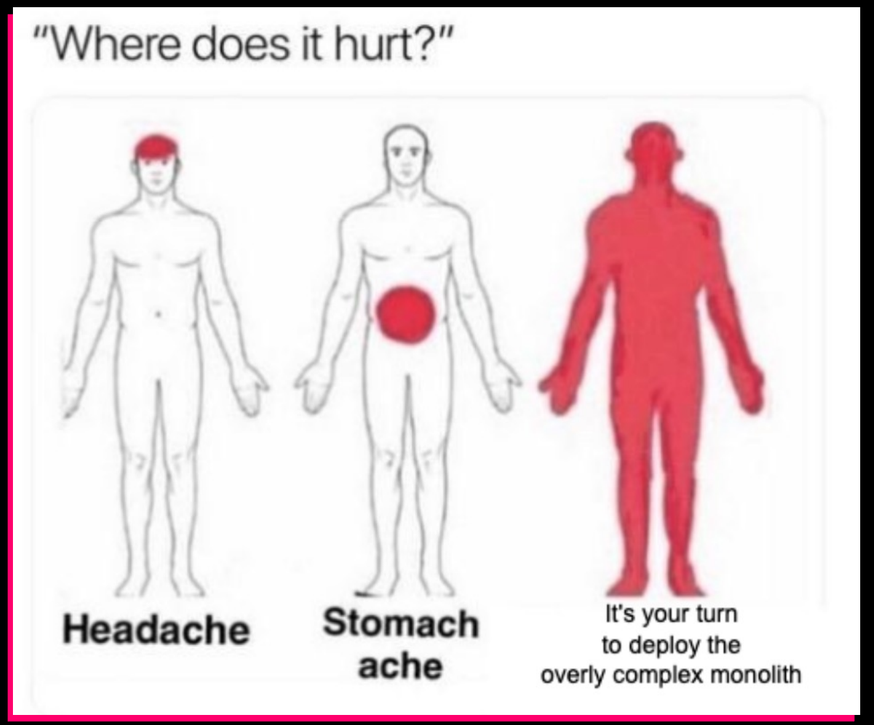
1. Audit
2. Reverse and Requirements Engineering
3. Phased Roadmap
4. Defining a MACH-aligned Architecture
5. System Overview
6. Shopware to MACH
7. Key Trade-Offs and Lessons Learned

1

Audit

1. Audit

- What we found: complex or at least feature-rich monolith
- But why did this hurt to manage?
  - More budget consumed for maintenance than for further development
  - Deployments always seemed daunting and somehow have been risky
  - Issues occurred that directly impacted revenue
  - …



… overgrown monolith with
excessive complexity,
no single source of truth,
and a talent for burning budget

1. Audit

- Defining the North Star:
  - More features, less maintenance
  - Become the go-to shop in the segment
  - Expansion
- Align stakeholder business goals with architectural requirements

2

Reverse and Requirements Engineering

2. Reverse Engineering

- Shopware system as multi-talent included:
  - Full ETL processes to sync data from and into an ERP system
  - Smartphone app integration causing high loads
  - Lots of asynchronous stuff like generating custom variant logic
  - …

We had to get to know what each of this processes does.

2. Requirements Engineering

- Look through business processes
- Understand stakeholder and customer needs
  - Audits
  - Persona + Customer journey
- Define features and split them
  - What’s needed to sell?
  - What’s needed to become better?

3

Phased Roadmap

3. Roadmap and MVP

Translating requirements into a phased roadmap with two phases:

Phase 1: The MVP – Reduced and scalable status quo

- Stripping back to core features and urgent customer needs
- Establishing a composable core

3. Roadmap and MVP

Phase 2: Value and differentiation – Innovation leap

- Features that move the needle
- Try out experimental features

4

Defining a MACH-aligned Architecture

4. What is MACH and Composable Commerce?

Microservices, **A**PI-first, **C**loud Native, **H**eadless Frontend

Beyond the buzz:

1. Divide systems into disparate applications with single responsibilities
2. Develop them in a way that allows for continuous delivery
3. Increase flexibility and resiliency to change

   by designing contracts

4. Opportunities & Risks

Opportunities:

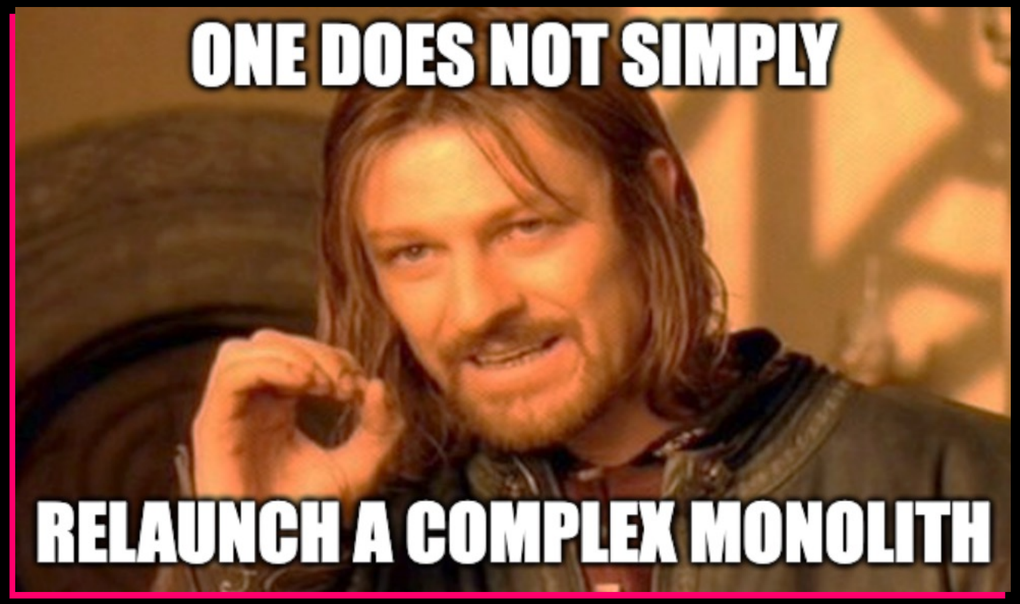
- Clear separation of concerns, layered failure mode
- Introduction of new concerns is easy and low in risk
- Allows parts of the system to grow at different velocities

Risks:

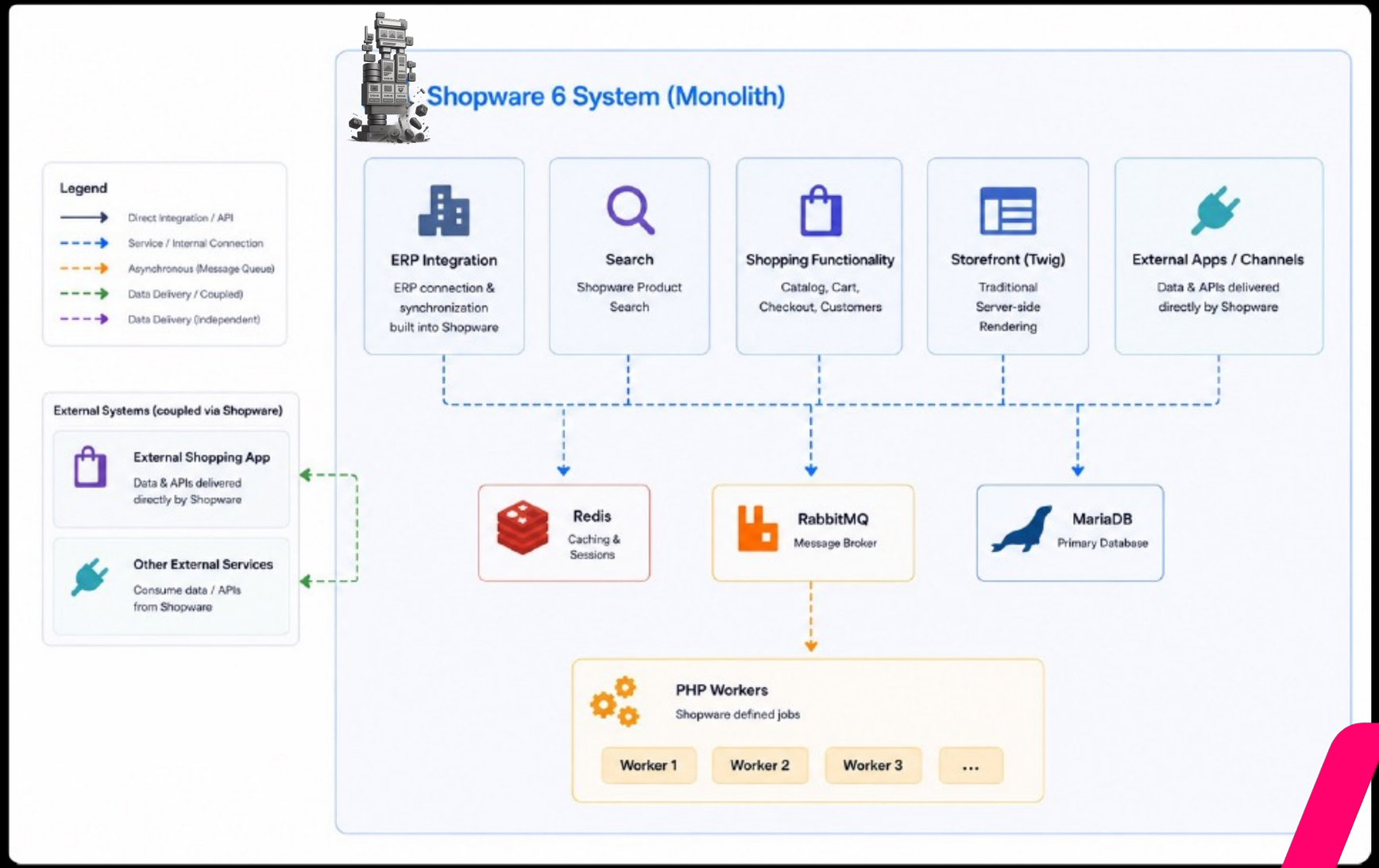
- Higher complexity: More moving parts, more deployments
- Lowered fault tolerance compared to monoliths
- High automation effort to increase efficiency

5

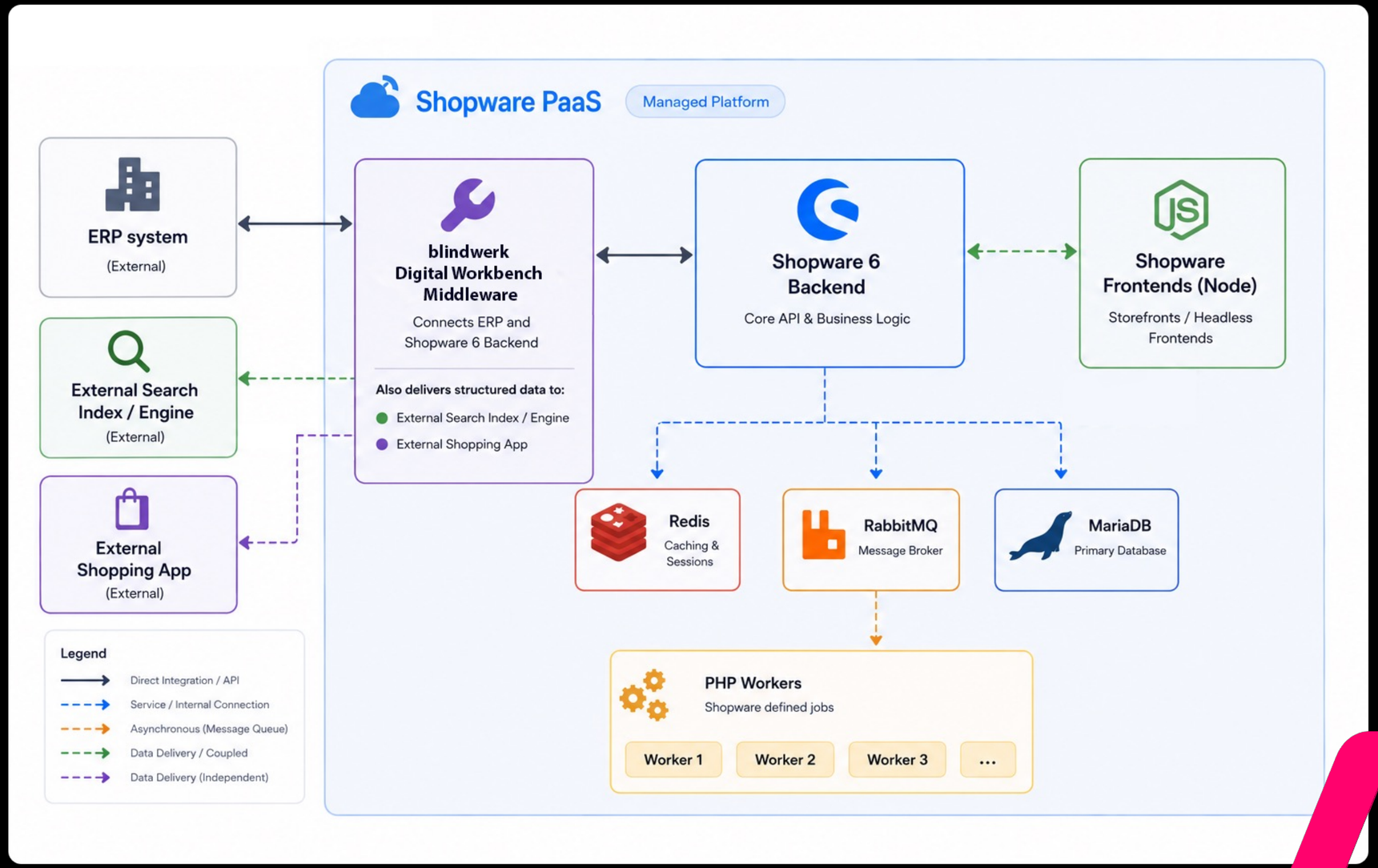
System Overview


ONE DOES NOT SIMPLY
RELAUNCH A COMPLEX MONOLITH

5. System Overview (before)



5. System Overview (after)



6

Shopware to MACH

6. Shopware to MACH

- **M**icroservices
  - Moved and reorganized “non-shop” functionality
  - Shopware backend as performant commerce engine
  - External / add-on features reflect in separate services
- **A**
- **C**
- **H**

6. Shopware to MACH

- **M**
- **A**PI-first
  - Shopware’s fundamental design: Store, Admin and Sync API
  - Make other service functionality available by APIs where necessary
- **C**
- **H**

6. Shopware to MACH

- **M**
- **A**
- **C**loud native
  - Shopware PaaS offers fully managed environments (Upsun Grid hosting)
  - Services are run in Nix-Runtimes (comparable to containers)
  - Scale only where necessary (not whole monolith)
  - Environments can be reached by Shopware PaaS-CLI
- **H**

6. Shopware to MACH

- **M**
- **A**
- **C**
- **H**eadless Frontend
  - Shopware Frontends (SDK) powered by Nuxt and Vue
  - Standard commerce logic is integrated as Vue Composables
  - Easily connect to various data sources
  - Swap data sources at will

7

Key Trade-Offs & Lessons Learned

MACH to rule them all?


MACH TO RULE THEM ALL?
NO

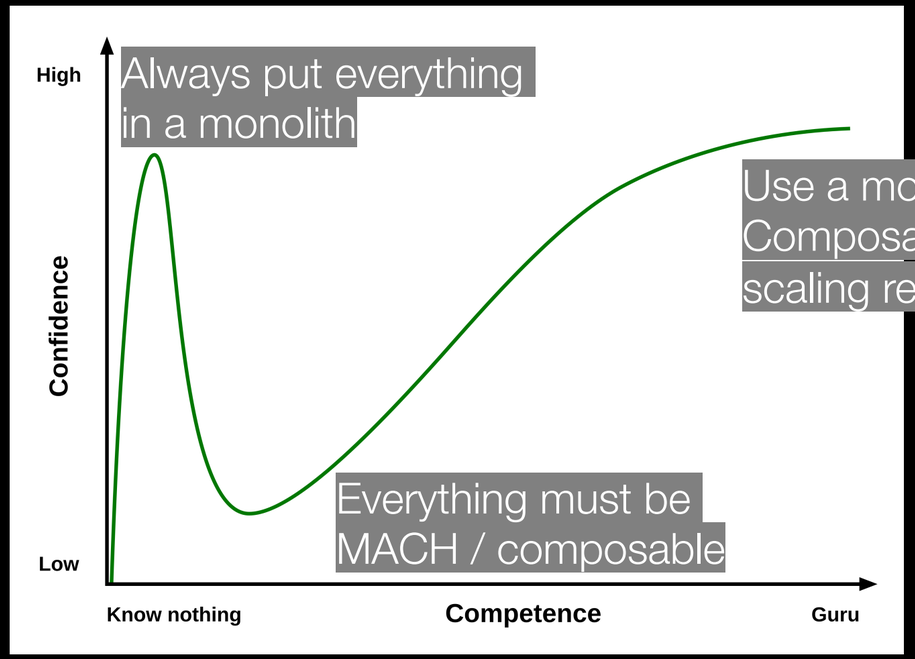
7. Trade-Offs & Lessons Learned

- Composable Commerce is no magic wand or all-purpose remedy
- Distributes complexity and makes it manageable
- Saved future viability in our case
  - Multiple interfaces moving data
  - Decoupled frontend processes

    and data sources
  - …

7. Trade-Offs & Lessons Learned

- More and new technologies led to increased on-boarding time
- Distributed systems also mean more complex testing (e.g. integration) and maintenance



The key decision was not to just relaunch a shop, but to restructure processes, responsibilities and technologies into a scalable future.

blindwerk
digitale architekten

blindwerk
Digitale Architekten

anfrage@blindwerk.de
+49 6323-988529