Designing an architecture to reduce complexity, extend automation and deliver innovative digital products faster and better

Modularization and Cloud

GRAB’N GO 11 OCTOBER
The foundation for agility and innovation
Speakers

Benjamin Vogensen
Senior Manager, Systems Engineering
Deloitte Denmark
Agile Advisor

Stefan Grinsted
Director, Deloitte Digital
Deloitte Denmark
Architecture Advisor
Increasing complexity can limit the ability to deliver value fast enough. Organizations often need to manage multiple complex and interdependent development activities that are hard to govern, which would require them to rethink their architectural design.

### Need
- Meeting delivery criteria to changing needs
- Continuously deliver new functionalities

### Complexity
- Managing complex interdependent development activities
- Internal dependencies create problems and are difficult to govern

### Solution
- Needing to rethink approach to deliver new functionalities
- Utilizing modularized cloud technology to deliver a composable architecture
Becoming agile to deliver more innovation
How many organizations attempt their agile journey starting by transforming their processes

1. Initially, focus was on **improving and transforming processes**
2. Then the **organizational transformation** was in focus to ensure **dedicated product teams**
3. Now, we see focus on **architecture** as an **enabler for independent teams and effective processes**

Processes | Organization | Architecture
Experience shows that by first focusing on building the architecture correctly will act as enablement for better organized product teams, more effective processes and support Project to Product transitions.
From a monolithic architecture to a modularized architecture
A modularized architecture can “open the door” for more innovation and ingenuity

A modular system is a collection of building blocks that can be configured in different ways, adapting for different customer needs.

Standardized modules that each implement one or a few functional elements in their entirety can be put together to continuously configure and evolve solutions to customers.
How Domain-Driven Design can help in this journey
A good old approach made increasingly popular by emerging technologies

- Originally outlined by Eric Evans in the seminal book “Domain-Driven Design: Tackling Complexity in the Heart of Software”
- A way to model complex software architectures long before microservices
- Build Ubiquitous Language that embeds into the Software
- Strategic Design - organize large domains into a network of Bounded Contexts
The journey to a modularized architecture
And get organized around domains and capabilities

0. Starting from Business capabilities
1. Grouping capabilities into Domains
2. Identify the Bounded Contexts
3. Define Product Architecture
4. Reorganize based on architecture
5. Iterate and Implement

Capability map
Domain Map
Context Map
Target Architecture defined
Product Teams established and kicks off
Continuous improvement
Modularization is a prerequisite to composable
What being composable adds on top of a modular architecture

- **Autonomous**
  Changes in a module don't impact other modules.

- **Orchestration**
  Modules have an agreed way of interacting

- **Discovery**
  Other teams can easily find what they need

- **Business Agility**
  Fast-acting, empowered and independent teams
Cloud is an enabler for a modularized and composable architecture
Utilizing cloud benefits to enable innovation, automation and engagement

1. **Makes Infrastructure Faster**
   Make changes on demand. This allows more efficient use of resources scale up or down and save cost, deliver services faster and with better quality.

2. **Utilizes APIs Quickly**
   When new services are available in the Cloud, it creates constantly new opportunities and ideas for products and functionality.

3. **Increased Automation**
   Automation is the way of the future and companies that automate internal processes have quicker service and a higher level of internal organization and collaboration.

4. **Team Engagement**
   Enable business and technology teams to be more integrated and engaged in delivery.
How Cloud can Support Modularization
The many different cloud capabilities that can help build and maintain a modular architecture

- Containerization services
- Serverless services ecosystem
- API Monitoring
- API Gateway
- Content Delivery Network
- DNS service
- Load Balancer
- Thread detection
- Security control
- Diagnostics and monitoring
- Security logging and audit

- Cloud File Storage
- Managed, relational and non-relational database services
- Cloud-native databases
- In memory caches
- Data migrations tools
- Development tools
- Code storage
- Continues Integration
- Continues deployment
- Queue service
- Notification service
- Message Broker services
Case Study
How a consumer goods company increased its business agility and delivered value faster
The Company’s Problems and Goals

The combination of increasing demand for change and business agility, and the growing complexity of the company’s IT systems and infrastructure, action was needed to realize more value faster.

**Problems**

With their **monolithic architecture** and **siloed IT organization**, the company faced **challenges** in their further development:

- Slow speed-to-market
- Increasing Complexity
- Tight Coupling and Vendor lock-in

**Goals**

Becoming more **agile as a business** in order to **respond faster** to market trends and customer needs through:

- A domain- and capability centric organization
- With independent and empowered product teams
- Using a shared language
Enabling Domain-Driven Design by Architecture
How the company’s domain-driven design philosophy and modularization synergize to reach the goal

Domain-driven design (DDD)*

- Build a **Ubiquitous Language** *(Shared Language)* and embed it
- Organize domains into **Bounded Contexts** *(Functional Areas)*
- Foster collaboration between **technical**- and **domain experts**
- **Iteratively** refine the conceptual model that addresses domain problems

Modularization on product level

Organize by **domains** and their **bounded contexts** to **modularize** each unique product

---

*Domain-driven design was coined by Eric Evans in his 2003 book “Domain-Driven Design: Tackling Complexity in the Heart of Software”*
Empowering Product Teams to be independent
Agreeing on a Published Language (interfaces), enables each Product to be isolated and composable

Domain

- Product Bounded Context
  - Domain Expert(s)
  - Product Team

Team decides how to deliver the business needs and e.g. to build vs. buy
The team owns and manages the full product life cycle

Other Domain

Provide a common authorized interface for other products using ubiquitous language
Utilizing Cloud to enable business agility through faster value-delivery

The company uses Cloud technology to ensure an intelligent and modular architecture

Using cloud **microservices** to create products

- Teams spend time on solving the business problem, not IT problems
- Teams synergize using similar technology stacks across products
- Infrastructure is faster than ever to spin up and is easily managed
The valuable benefits of a modularized and composable cloud architecture
Enables the required agility and innovation to constantly respond to emerging customer needs

- **Agility**
  Allows organizations to deliver new products, functions, and features more quickly and pivot more easily if needed

- **Innovation**
  Increased appetite for exploration, experimentation, risk-taking, and creative freedom enabled by quick hypothesis development and testing

- **Speed**
  Lower development and deployment time and thereby decrease in time-to-market for value-adding functionality

- **Efficiency**
  Easier to build and maintain as it adds ease to the process of identifying and resolving the root cause of performance issues

- **Flexibility**
  Enables potentially endless combinations by “mix and match” modules to achieve mass customization and continuously deliver value

- **Cost savings**
  When aiming to be highly agile and innovative a modularized and composable cloud architecture is a good business case as it will incur lower costs of frequent change
Transitioning from Project to Product for managing digital solutions, becoming more **innovative** and delivering **more value** is best enabled by a modularized, composable and cloud-based architecture.