GEA Engineering for a better world.

NPCW 2026

Scaling a Process Control Product: Solution Architecture and Service Delivery in Industrial MPC

GEA Process Engineering A/S
17-08-2026

GEA OptiPartner

Digital Optimization Team

What do we do?

We make **model-based process control** autopilot solution for industrial scale processes, as a subscription-based software (SaaS)

What is GEA OptiPartner?

This includes **Real-Time Optimization** (RTO / *Local optimization*) and **Model Predictive Control** (MPC / *Supervisory control*) for a given process. The subscription includes **Performance Care** and continuous updates for the software

What Processes do we Support?

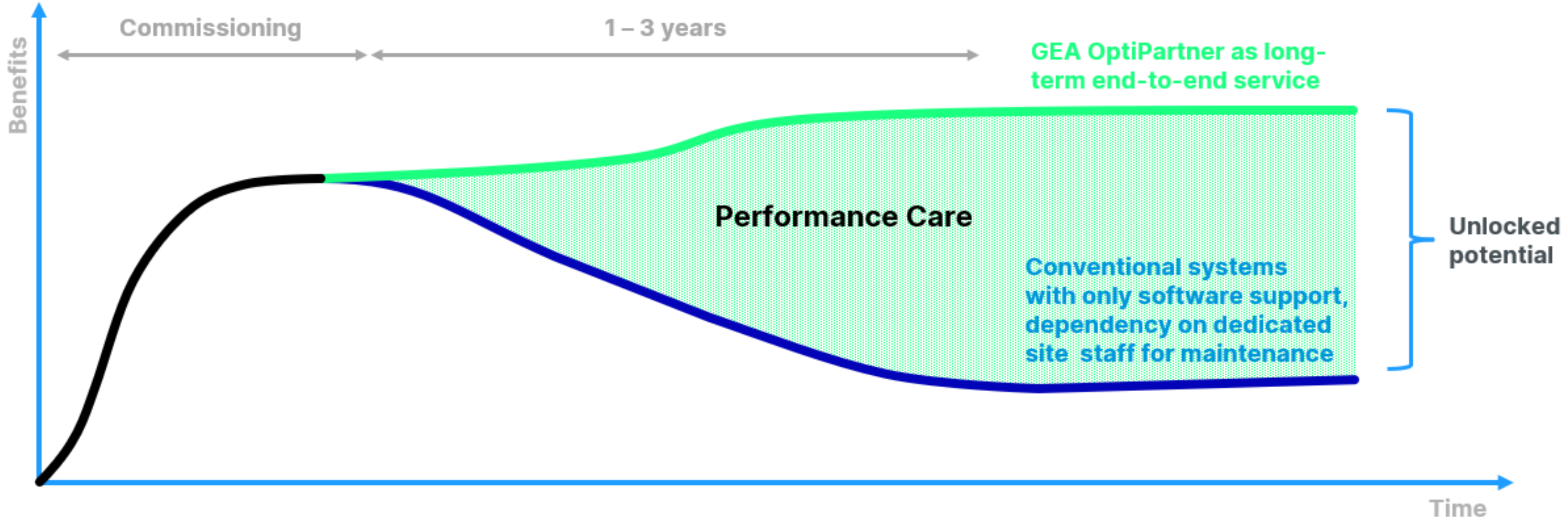
Large install-base across industrial processes including;
Spray Drying, **Freeze Drying**, **Falling Film Evaporation**, and **Membrane-filtration**

(*) Sped-up example of the software running at a customer site

Process ⟷ **Regulatory Control** ⟷ **Supervisory Control** ⟷ **Local Optimization** ⟷ **Scheduling**

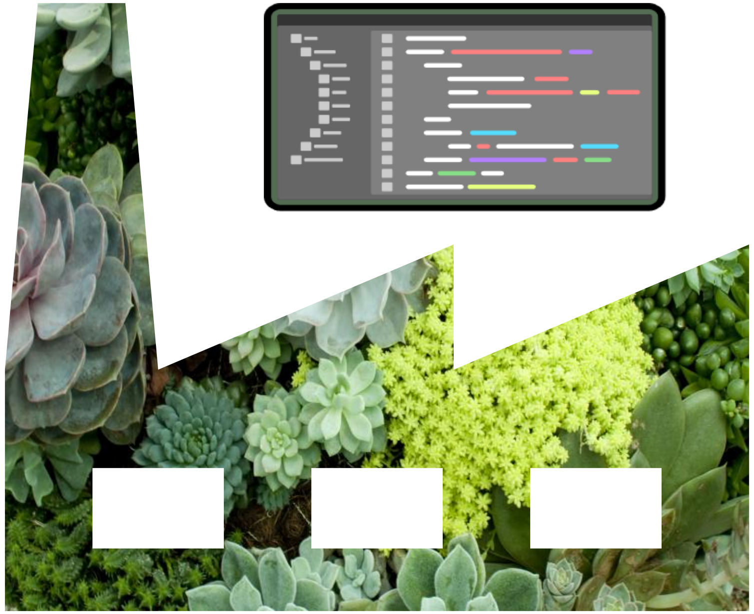
You need to take care of your plants

Optimization is a never-ending story



Remote operation meetings	System monitoring	Performance review meeting	Pro-active call for update

- Models, parameters and the underlying software needs to be maintained and cared for
- SaaS products are like house plants, if they are not watered/cared for they wither and die



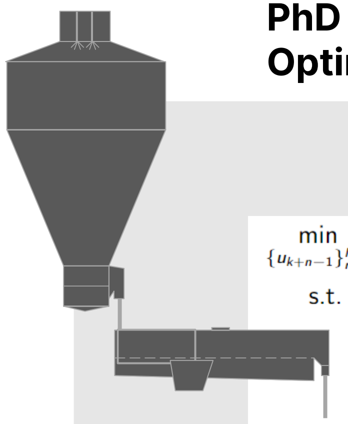
Sales Status for July 2026

GEA OptiPartner
75 Modules

Humble Beginnings

PhD project turned into a commercial MPC product

PhD Project in Economic Optimization of Spray Dryers



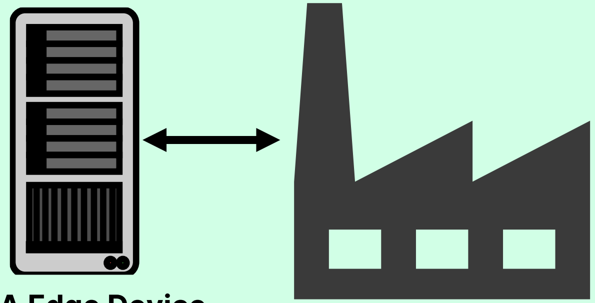
MATLAB

$$
\begin{aligned}
\min_{\{u_{k+n-1}\}_{n=1}^{N}} \quad & \phi \\
\text{s.t.} \quad & x_{k+n+1} = Ax_{k+n} + Bu_{k+n} + Ed_{k+n}, \\
& z_{k+n} = Cx_{k+n} + Du_{k+n} + Fd_{k+n}, \\
& z_{\min,k+n} \leq z_{k+n} \leq z_{\max,k+n}, \\
& u_{\min} \leq u_{k+n} \leq u_{\max},
\end{aligned}
$$

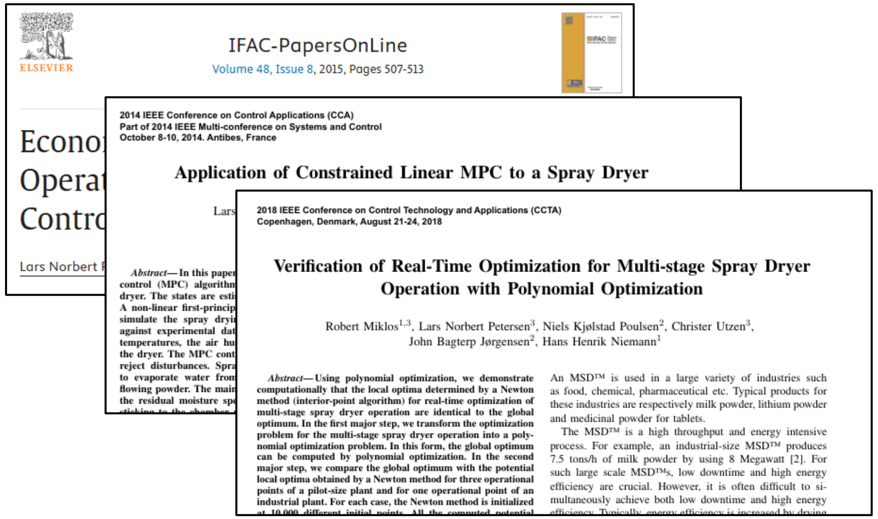
Deployed on-site

(Edge/On-Premise)

Partner Company



GEA Edge Device



ELSEVIER

IFAC-PapersOnLine

Volume 48, Issue 8, 2015, Pages 507-513

Econo… Operat… Contro…

Lars Norbert P…

2014 IEEE Conference on Control Applications (CCA)
Part of 2014 IEEE Multi-conference on Systems and Control
October 8-10, 2014. Antibes, France

Application of Constrained Linear MPC to a Spray Dryer

2018 IEEE Conference on Control Technology and Applications (CCTA)
Copenhagen, Denmark, August 21-24, 2018

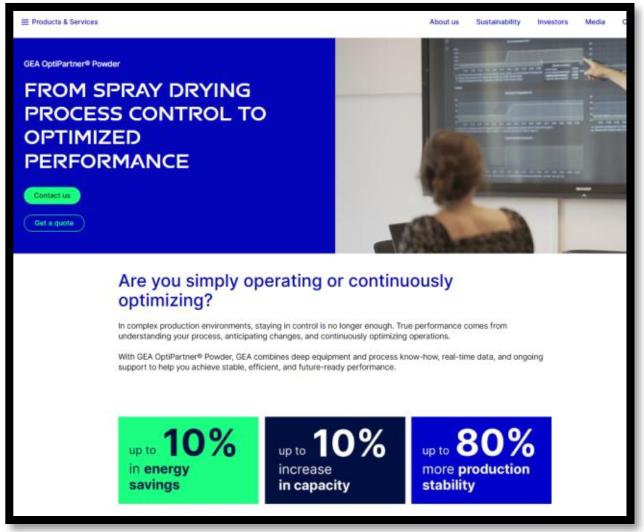
Verification of Real-Time Optimization for Multi-stage Spray Dryer Operation with Polynomial Optimization

Robert Miklos[1,3], Lars Norbert Petersen[3], Niels Kjølstad Poulsen[2], Christer Utzen[3], John Bagterp Jørgensen[2], Hans Henrik Niemann[1]

Abstract— Using polynomial optimization, we demonstrate computationally that the local optima determined by a Newton method (interior-point algorithm) for real-time optimization of multi-stage spray dryer operation are identical to the global optimum. In the first major step, we transform the optimization problem for the multi-stage spray dryer operation into a polynomial optimization problem. In this form, the global optimum can be computed by polynomial optimization. In the second major step, we compare the global optimum with the potential local optima obtained by a Newton method for three operational points of a pilot-size plant and for one operational point of an industrial plant. For each case, the Newton method is initialized …

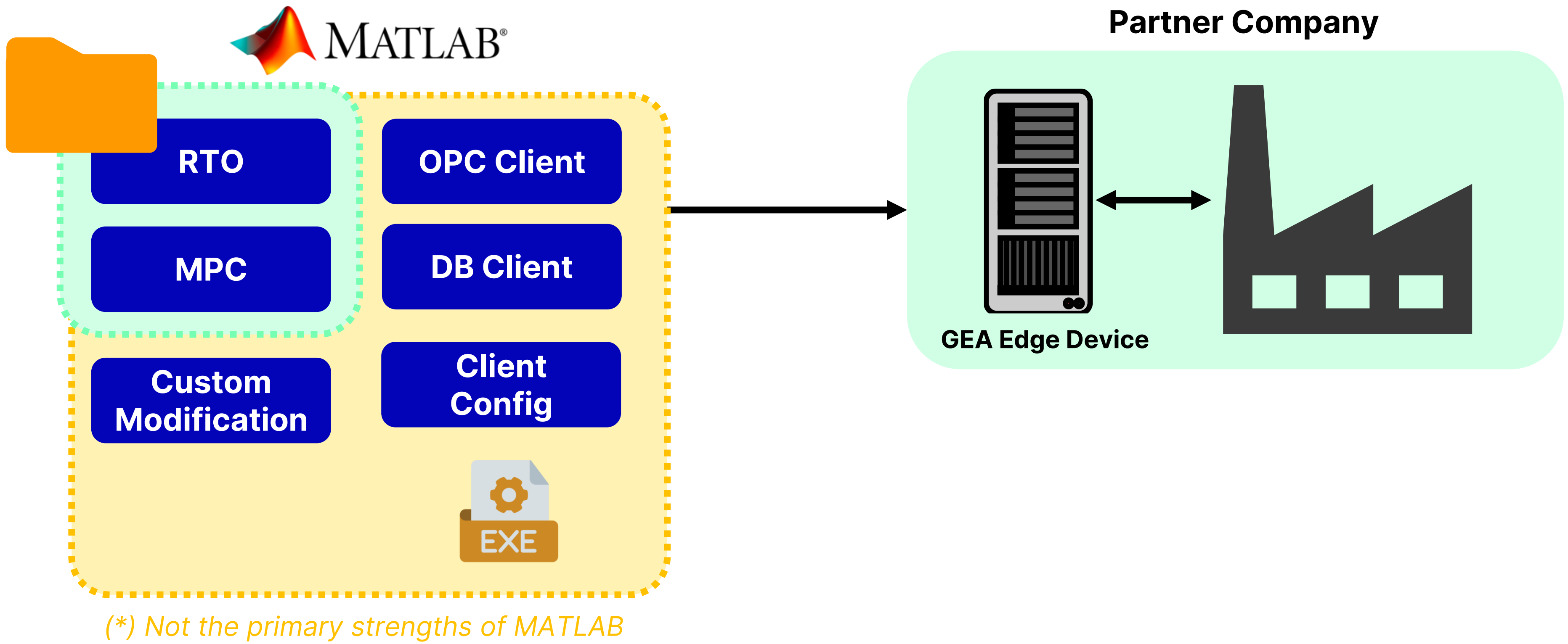
An MSD™ is used in a large variety of industries such as food, chemical, pharmaceutical etc. Typical products for these industries are respectively milk powder, lithium powder and medicinal powder for tablets.

The MSD™ is a high throughput and energy intensive process. For example, an industrial-size MSD™ produces 7.5 tons/h of milk powder by using 8 Megawatt [2]. For such large scale MSD™s, low downtime and high energy efficiency are crucial. However, it is often difficult to simultaneously achieve both low downtime and high energy …



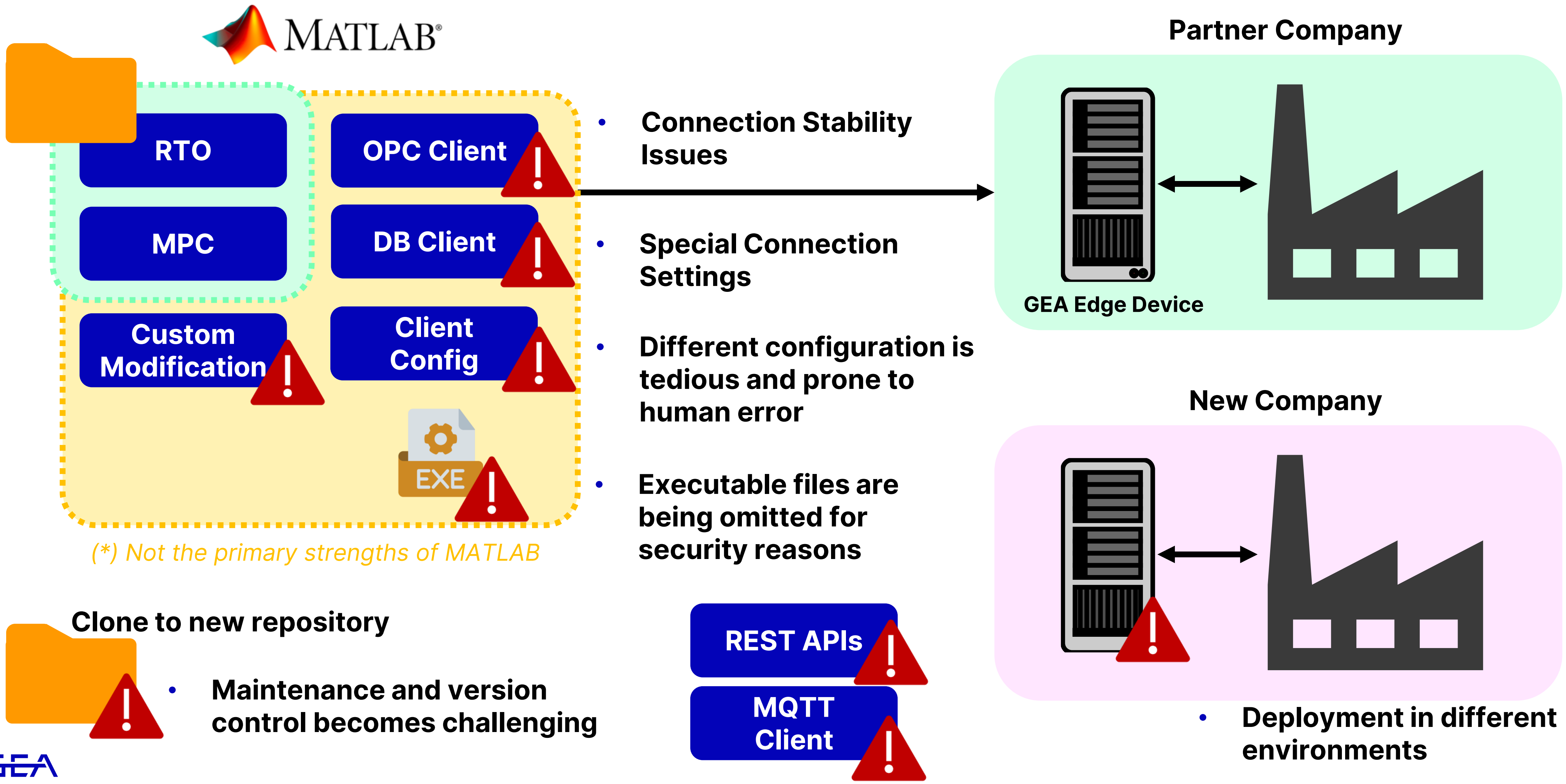
Practical Implementation

MPC & RTO requires auxiliary systems to work in practice



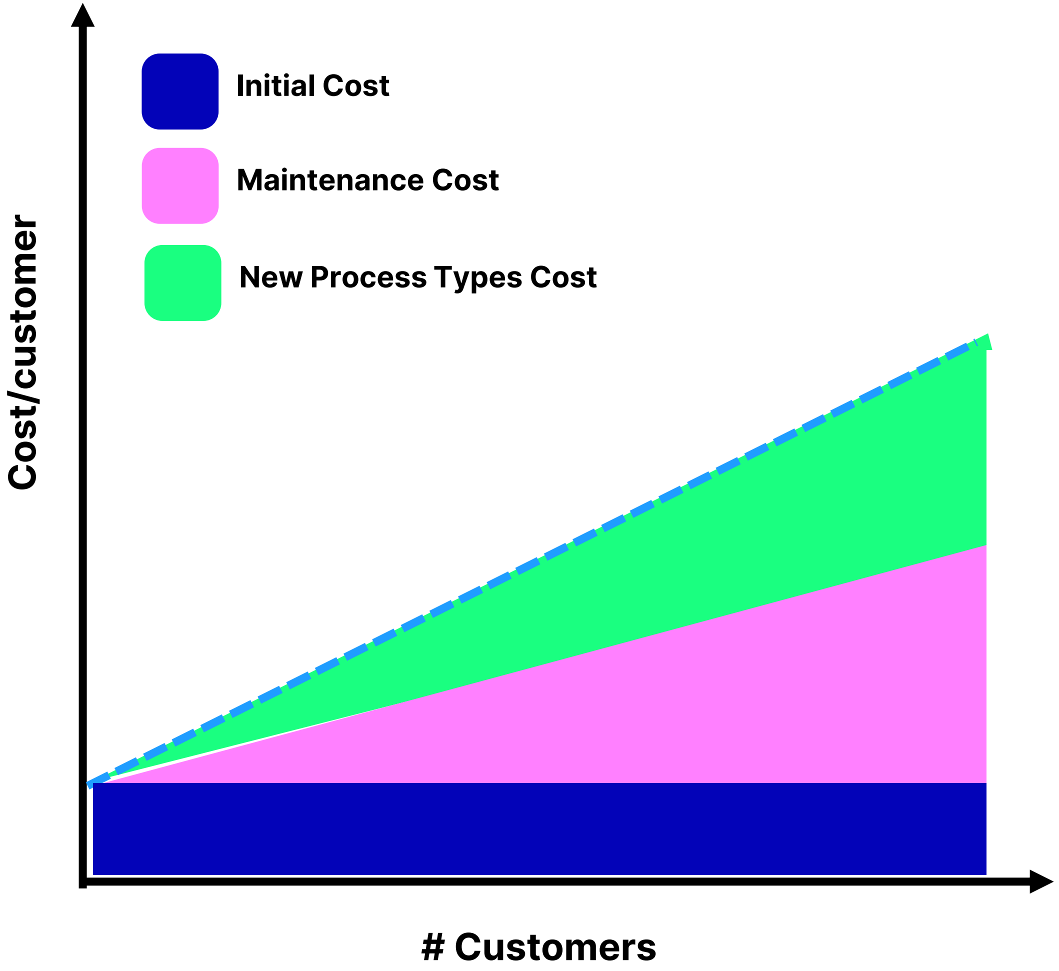
Challenges Encountered

The MATLAB stack showed challenges and limitations related to scaling



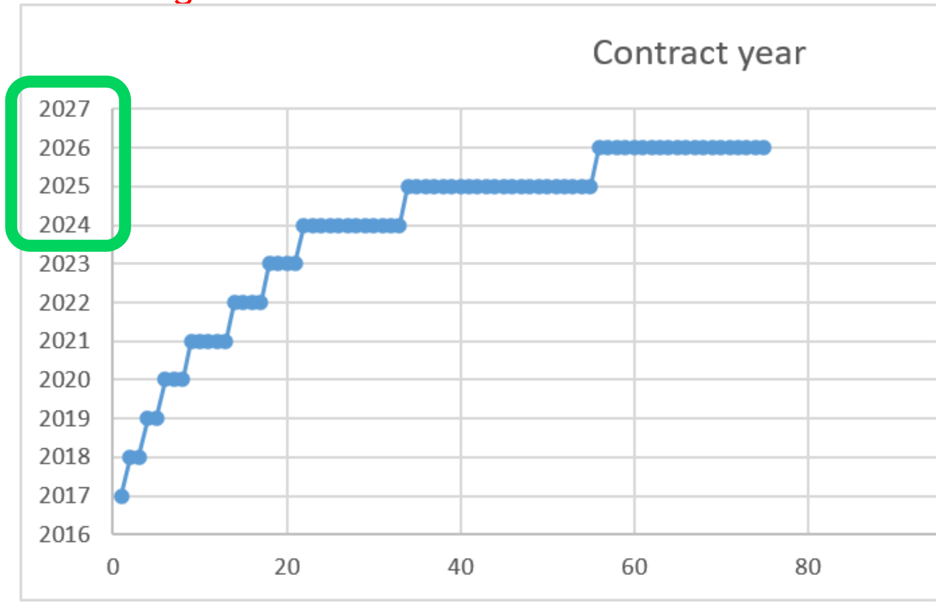
Challenges & Possible Solutions

Prioritize standardization and modularization



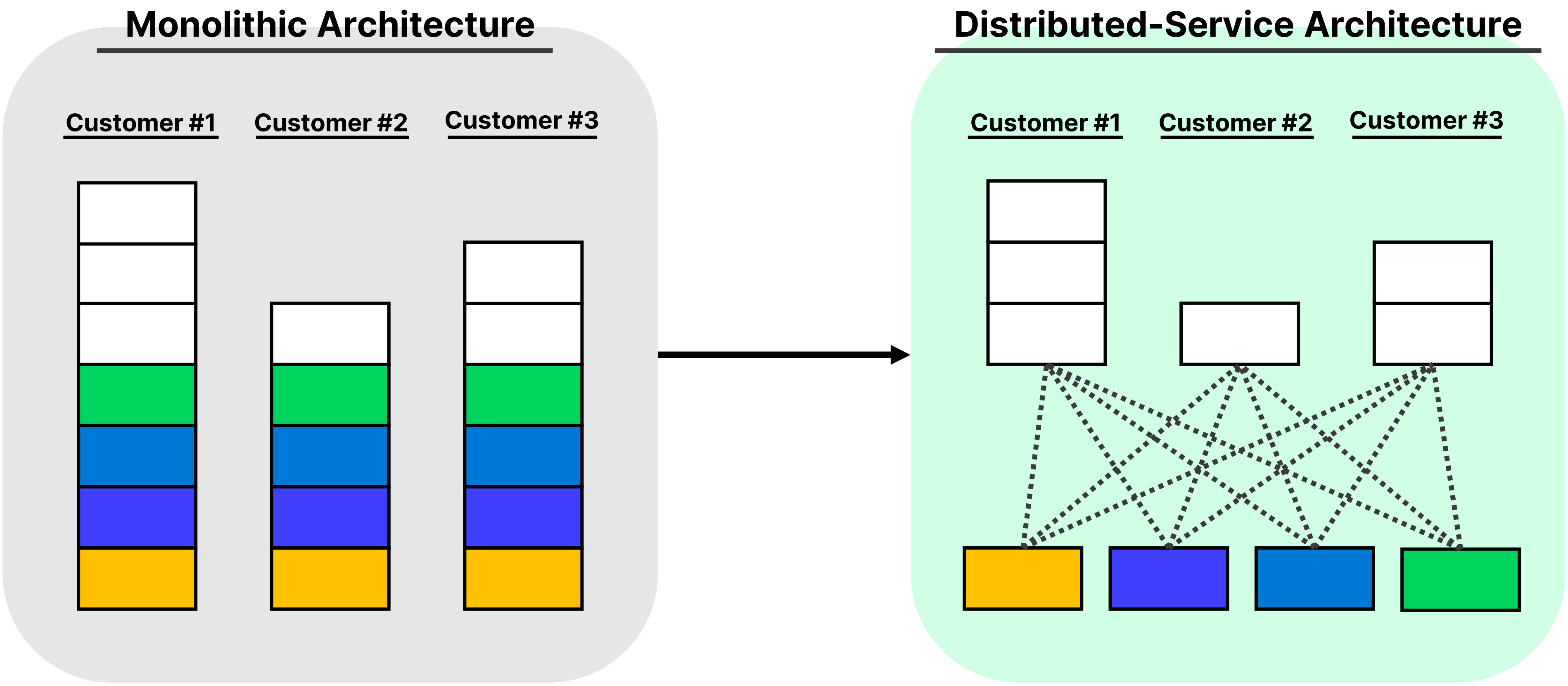
Monolithic to Distributed Services

- Separation of concerns
- Flexibility in setup (routing etc.)
- Reuse and ease of processes: Alignment of hosting and CI/CD (Docker & Build Server)
- Complexity in development, explicit/implicit contracts
- Complexity in communication, debugging and testing



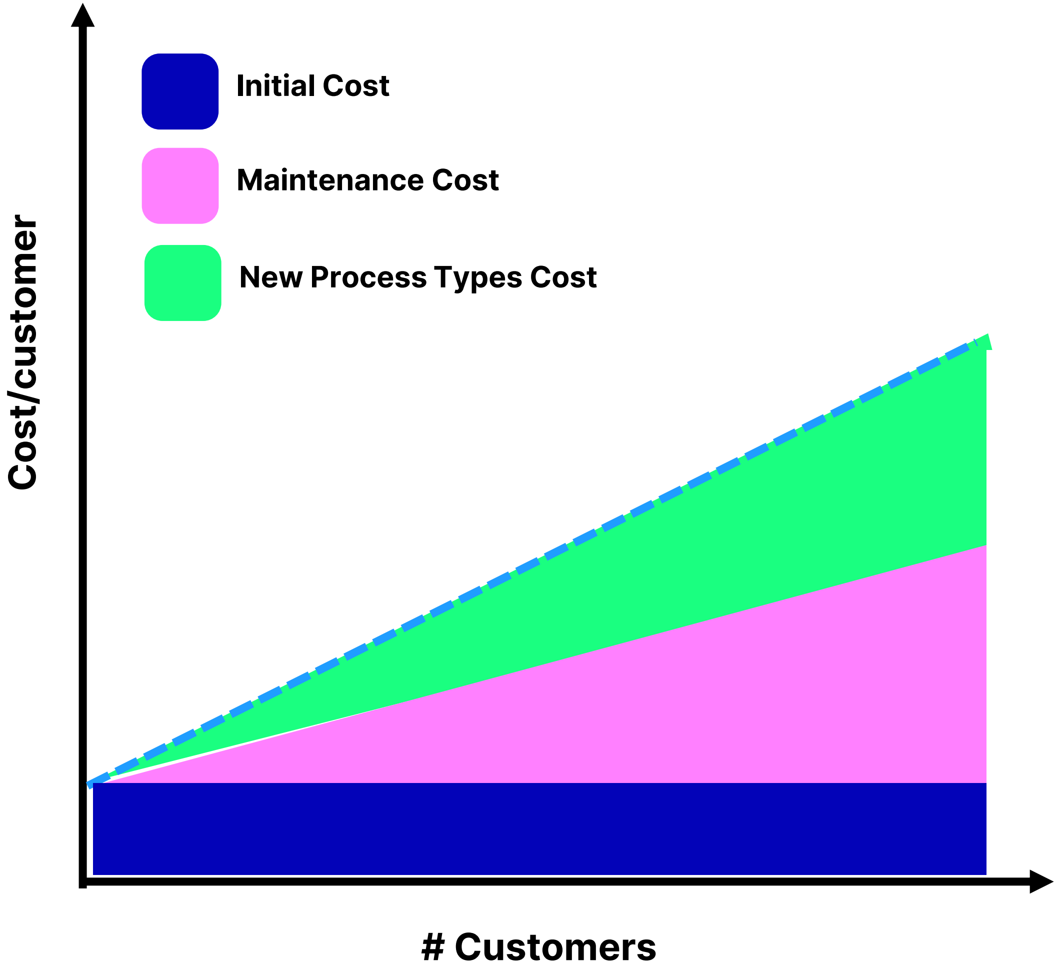
Comparison Between Monolithic & Distributed Services

Simplified example of the new application architecture



Challenges & Possible Solutions

Prioritize standardization and modularization



Monolithic to Distributed Services

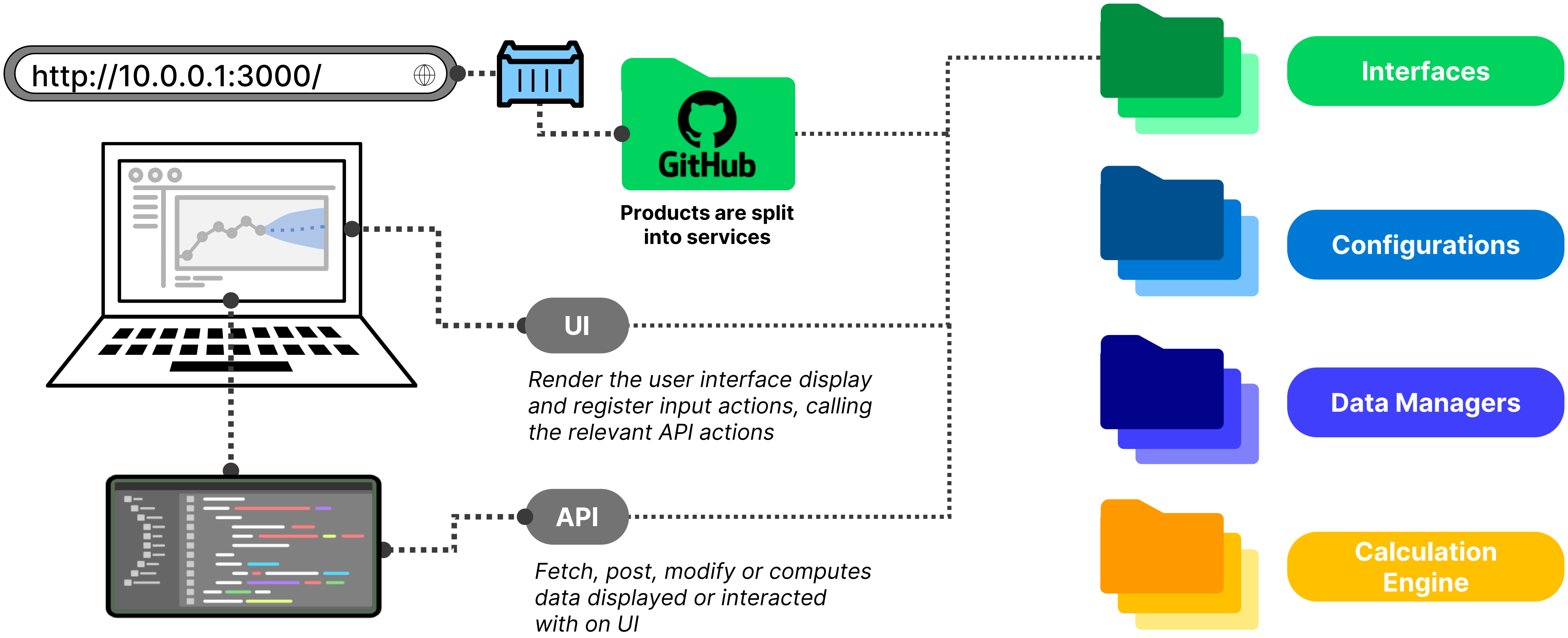
- Separation of concerns
- Flexibility in setup (routing etc.)
- Reuse and ease of processes: Alignment of hosting and CI/CD (Docker & Build Server)
- Complexity in development, explicit/implicit contracts
- Complexity in communication, debugging and testing

Standardisation & Version Control

- General improved QA
- Back tracking / improved debugging: Code, Configuration, and API/Interfaces
- Ensuring components compatibility
- Adds complexity to workflow

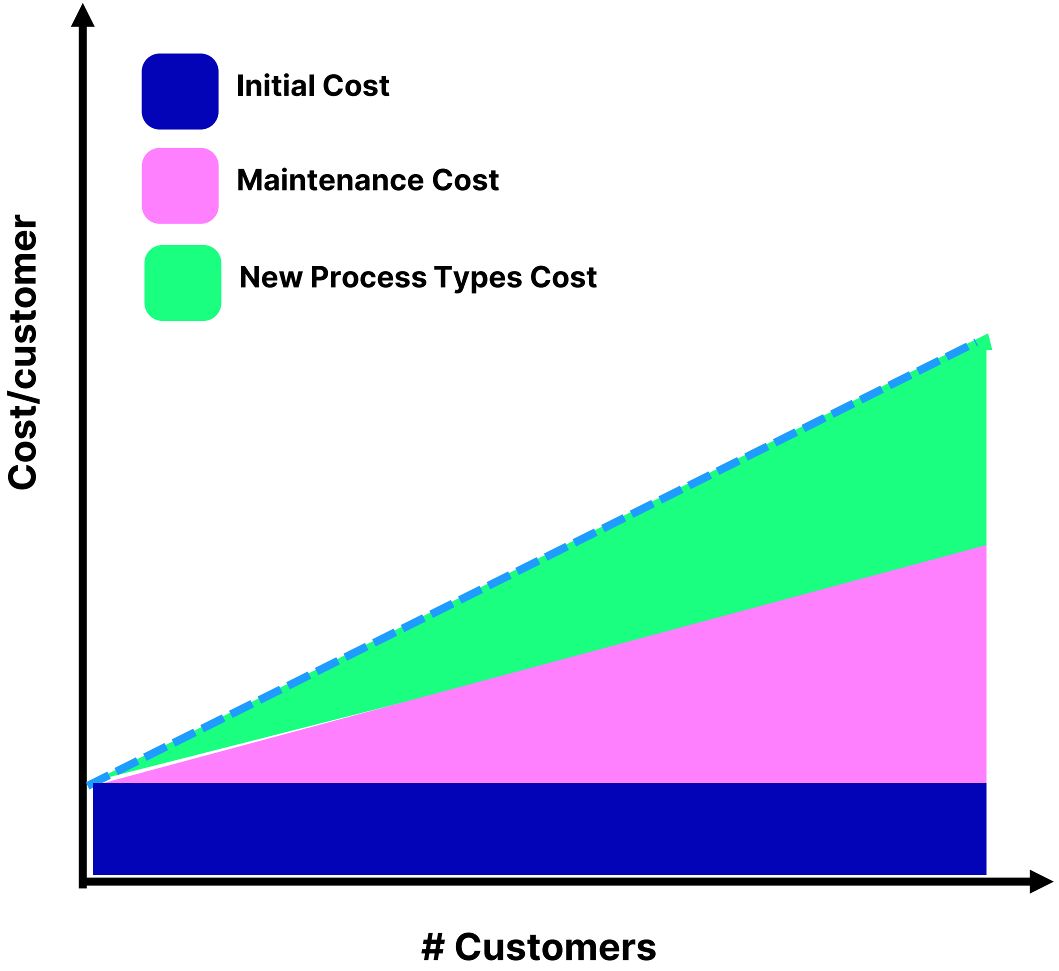
Distributed Service Application

Simplified example of the new application architecture



Challenges & Possible Solutions

Prioritize standardization and modularization



Monolithic to Distributed Services

- Separation of concerns
- Flexibility in setup (routing etc.)
- Reuse and ease of processes: Alignment of hosting and CI/CD (Docker & Build Server)
- Complexity in development, explicit/implicit contracts
- Complexity in communication, debugging and testing

Standardisation & Version Control

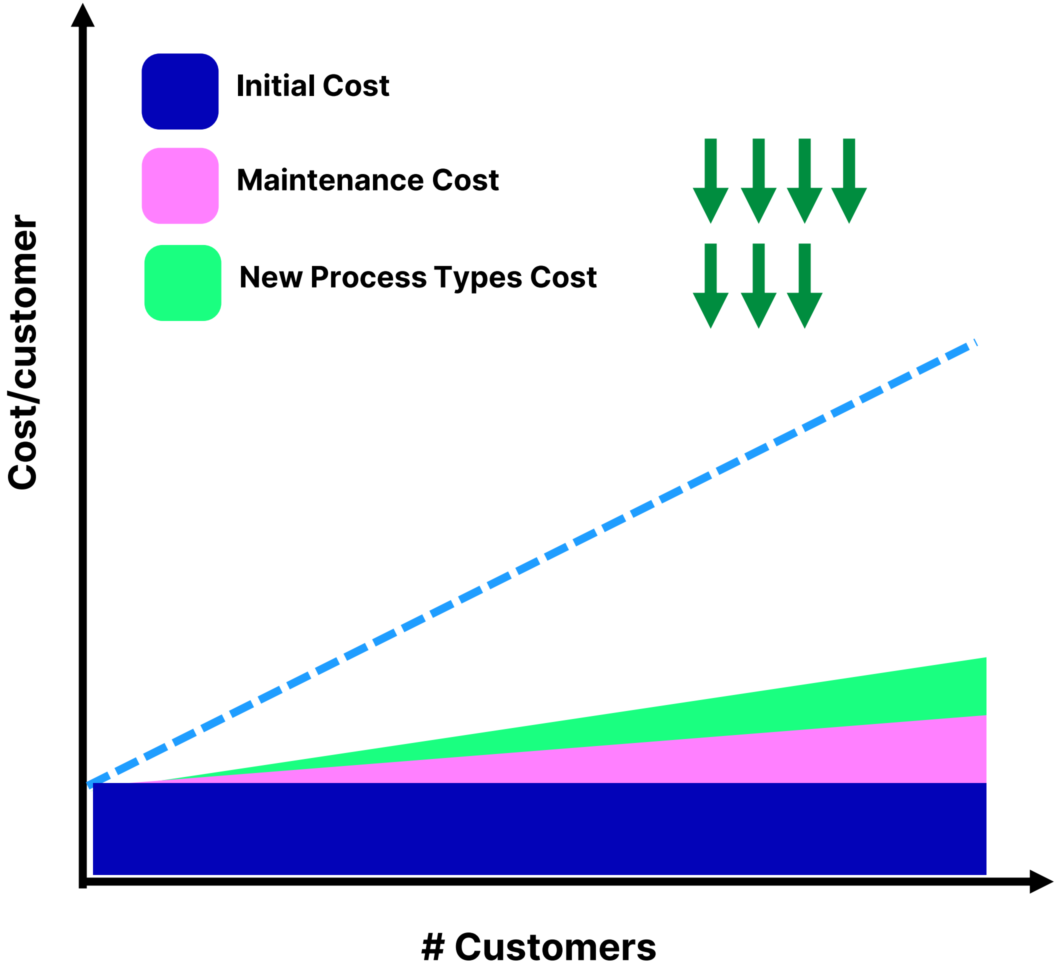
- General improved QA
- Back tracking / improved debugging: Code, Configuration, and API/Interfaces
- Ensuring components compatibility
- Adds complexity to workflow

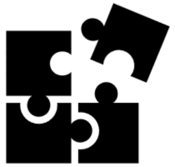
Translate from MATLAB to C#

- More stability in production environment (communication features)
- More code flexibility (cooperation, testing, benchmarking)
- Need to find numerical solver and math toolboxes

Observed Benefits of the Transformation

Prioritize standardization and modularization



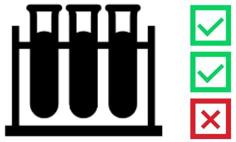


Standard implementation and new Interfaces

- Explicit dependencies makes the code predictable
- Easier to add new features that can be shared across other processes
- Easier to maintain

Modular architecture makes it easy to reuse code

- Containerized implementations allow for more holistically deployment strategies
- Adding new customers are now easier
- Less deployment friction

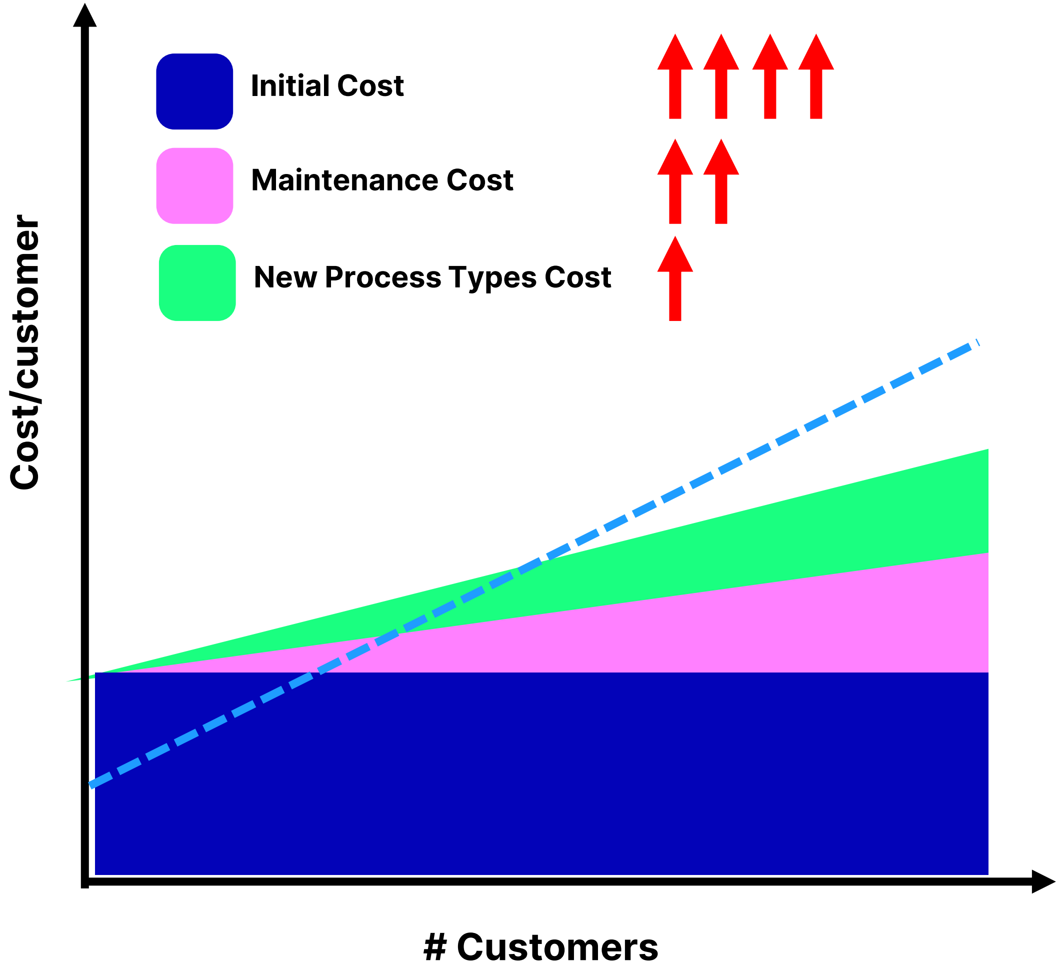


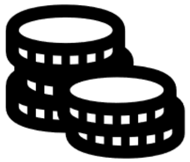
Testing and quality is easier in the modular setting

- Better quality tracking
- Catch bugs before production

Observed Drawbacks with the Transformation

Prioritize standardization and modularization



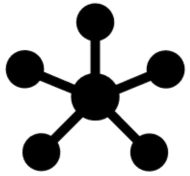


Higher initial cost due to development time

- The setup requires more planning and stricter adherence to standards
- Did not start from scratch, technical debt
- Overall, a longer initial development time

Requires different skill sets

- The required skill sets are more diverse
- It requires more people with different backgrounds
- Steep learning curve

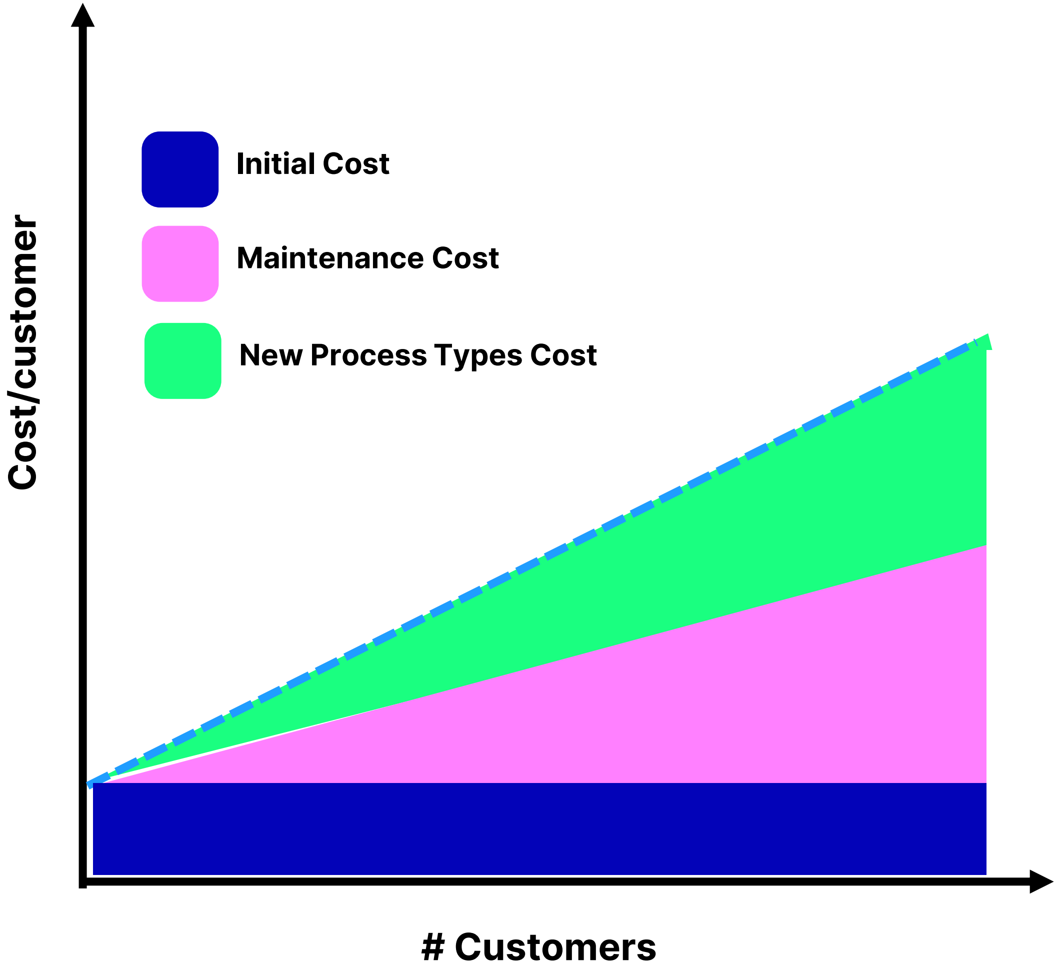


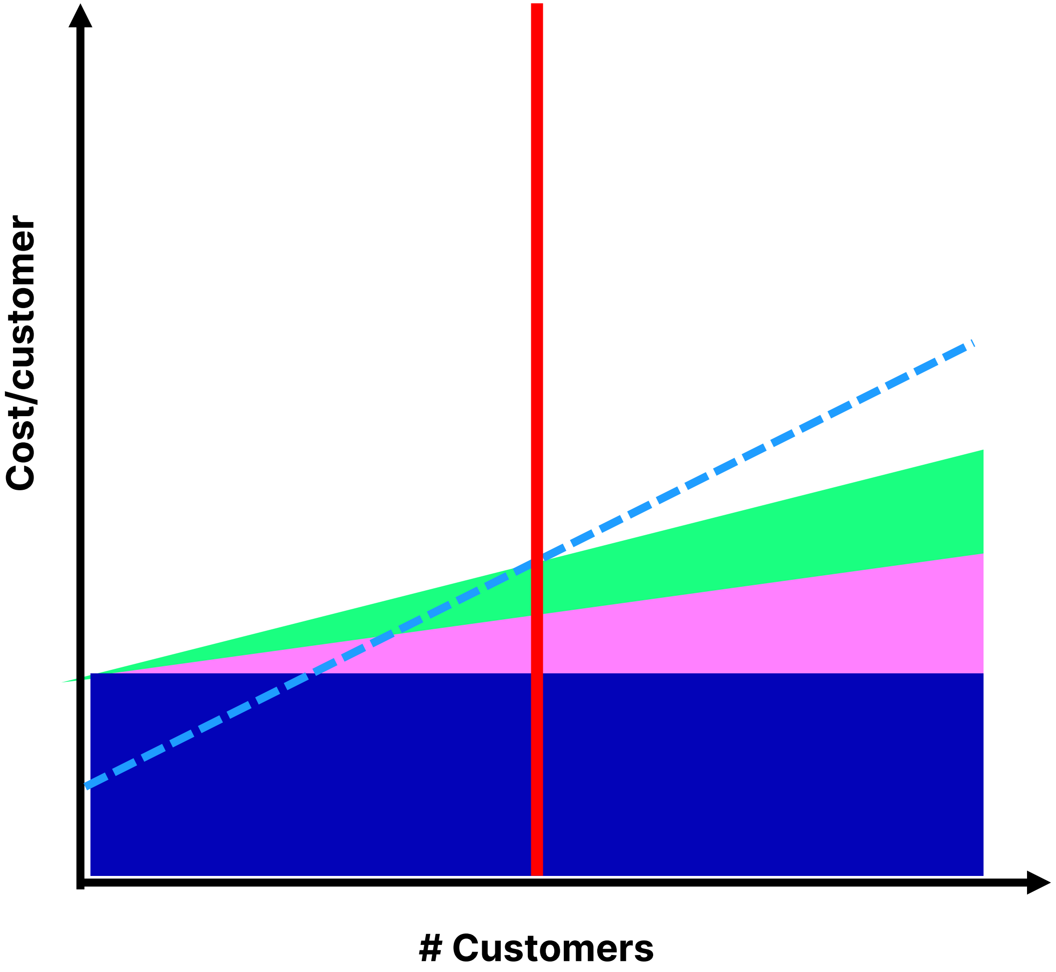
Connections and dependencies are more complex

- There were many initial set-up pitfalls that extend development time.
- Did make some development and implementation more tedious

When to do a bigger software transformation

There is a volume of customers where this becomes a benefit





Scaling Sales, Service & Support

Early Stage

- 1-5 customers
- Small team wearing many hats
- Ad-hoc analysis and diagnostic of live data

Growth

- 10-40 customer
- Dedicated roles & responsibilities
- Scripts for automated analysis and diagnostic of historic data

Scaled Operation

- +100 customer
- Standardized and shared knowledge transferred to dedicated responsibilities
- Dedicated diagnostic software (timeseries and logs), and proactive warnings

Summary & Conclusions

- **MPC is a service-based product, and it is never “complete”**
  - *Flexibility and maintenance is paramount*
  - *Adhering to standards (ISO) and modularity helped us*

- **Preemptive focus on scalability can be suboptimal**
  - *The MPC and RTO needs to work*
  - *Benefits are seen when you have difficulty meeting the customer demand*
  - *Initially, trust and customer-relation is more important*

You need to take care of your plants;
But don’t think about getting a watering system if you only have one

GEA Engineering for a better world.

THANK YOU FOR YOUR ATTENTION

ANY QUESTIONS?

Mads Stevnsborg
mads.stevnsborg@gea.com

Lukas Svane Theisen
lukas.svanetheisen@gea.com

Kevin Feldmann
kevin.feldmann@gea.com

Marcus Krogh Nielsen
marcuskrogh.nielsen@gea.com

Christer Utzen
christer.utzen@gea.com

Engineering
for a better
world.