FROM COMPILER SWITCHES TO STRATEGIC REUSE

Software Product Lines

Pär Hammarström 2015-10-16

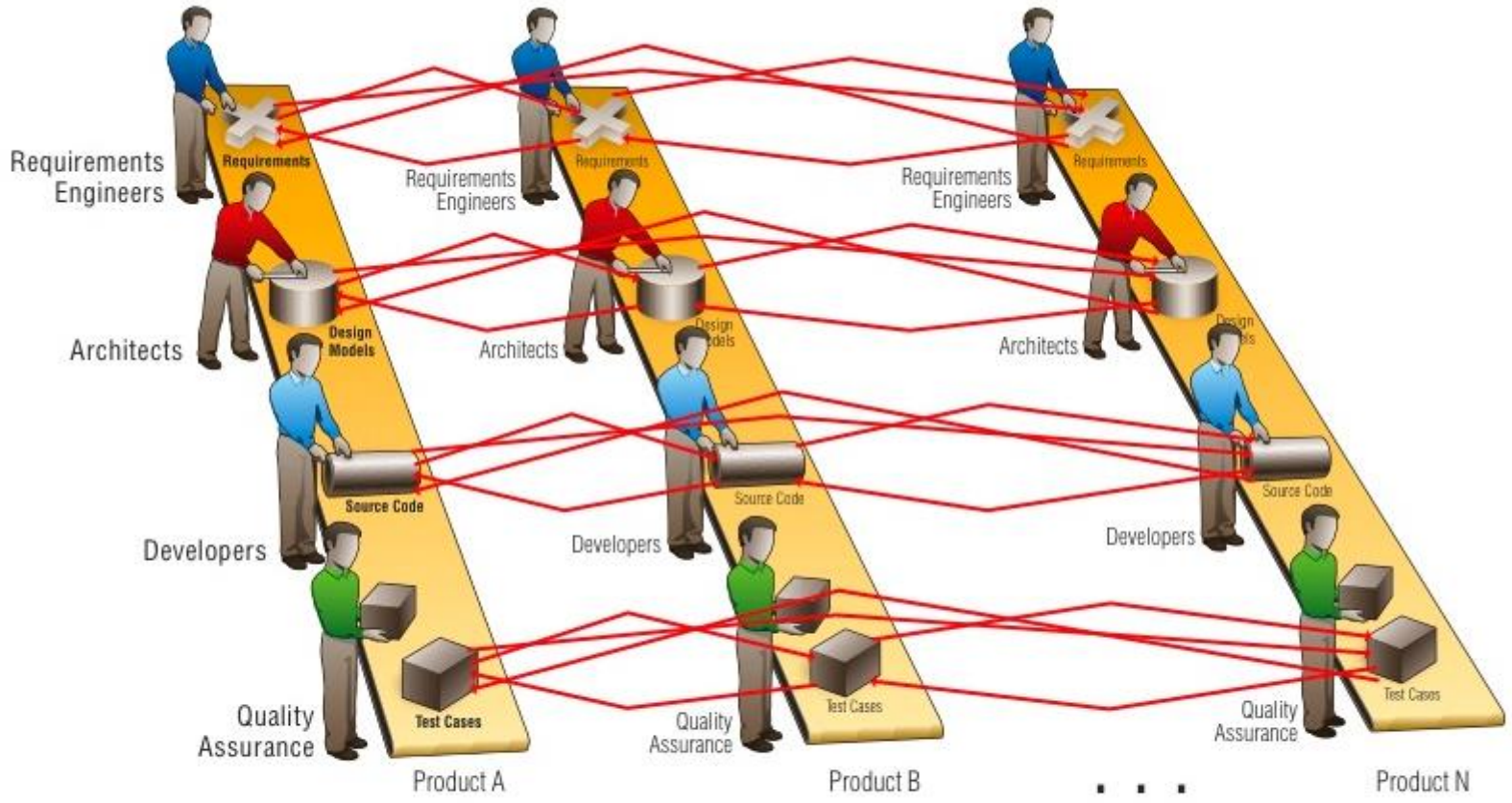
ABOUT ME

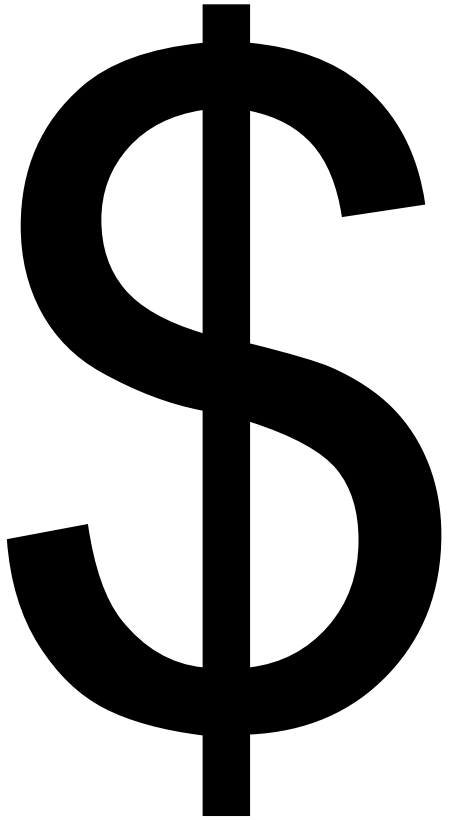
- Systems Architect and Development Manager
- 20+ years experience of Product Development
- Consultant, Development Manager, Product Manager
- Telecom, Automotive, ERP, Aerospace&Defence
- 4 in a row @Devlin

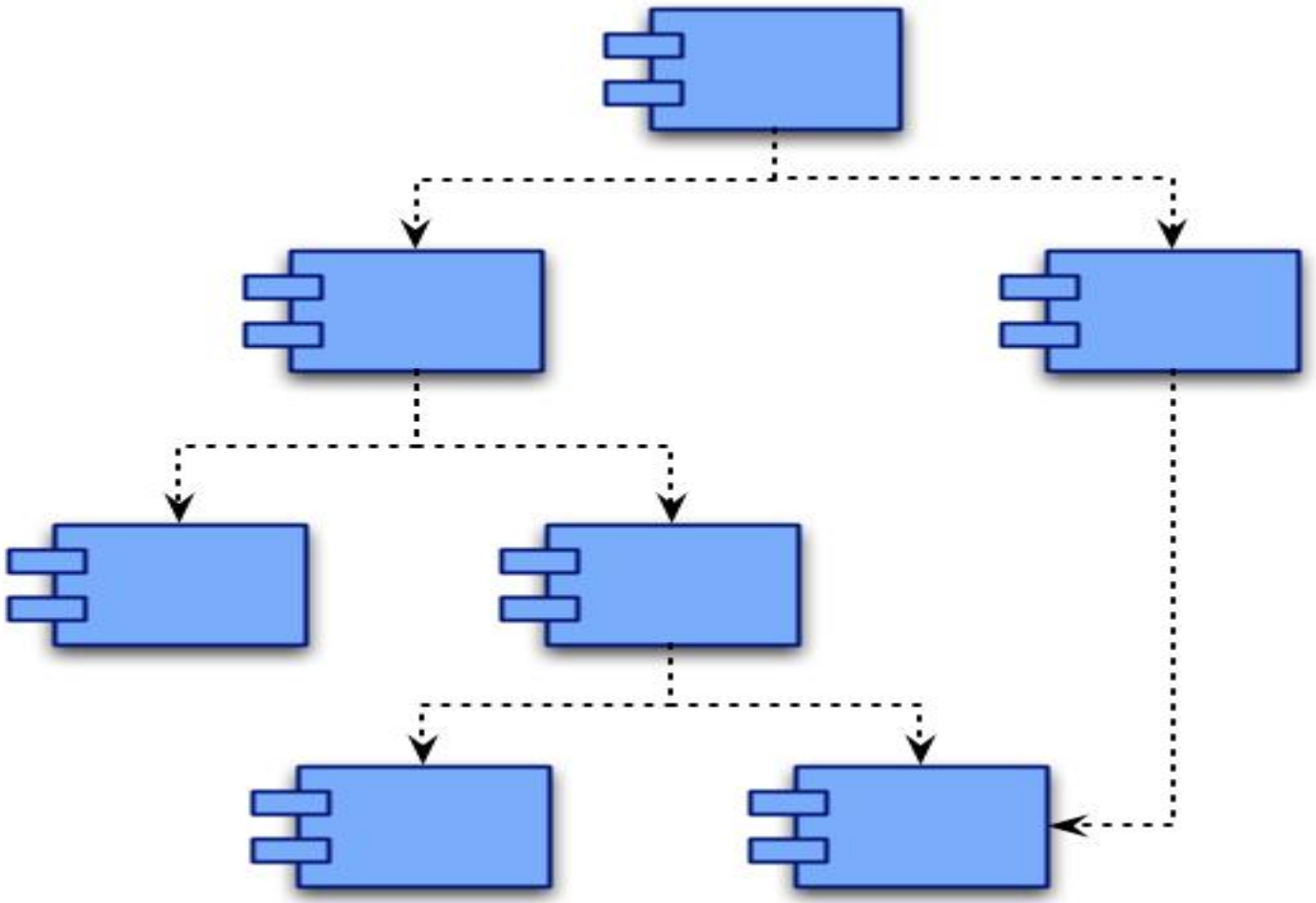
Pär Hammarström
SAAB Aeronautics
par.hammarstrom@saabgroup.com
073 418 0173

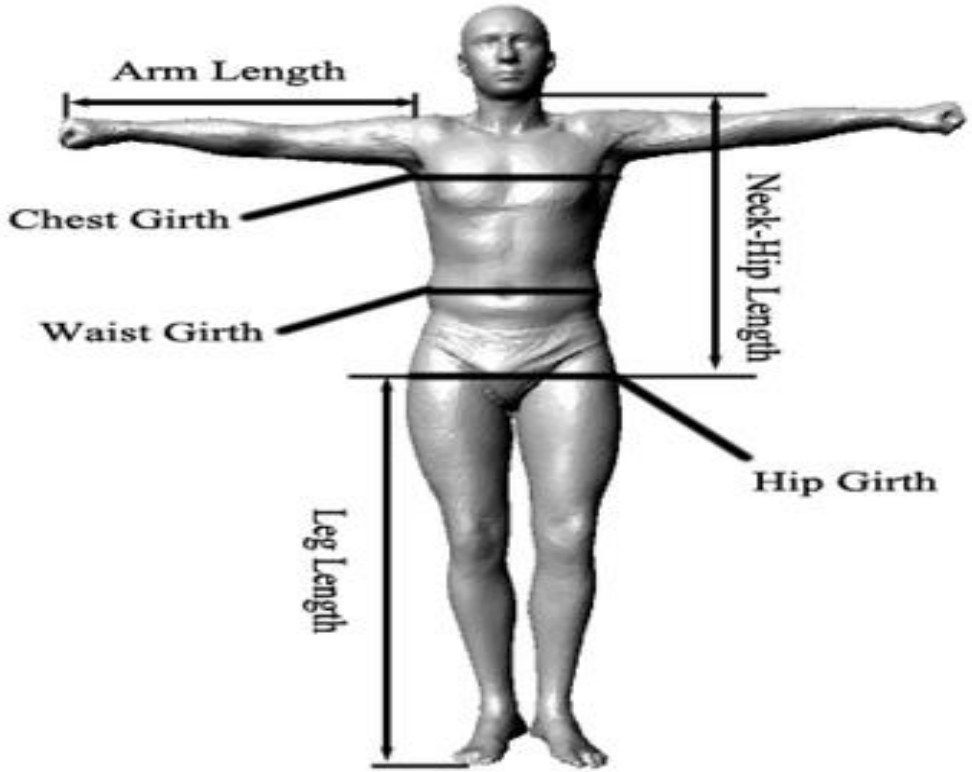
SOFTWARE AT WORK

SOFTWARE PRODUCT LINES








Arm Length
Chest Girth
Waist Girth
Neck-Hip Length
Hip Girth
Leg Length

(a)

#ifdef

```
#include <stdio.h>
int main()
{
   #ifdef PRODUCT_A
      char foo[] = “Welcome”;
   #else
      char foo[] = ”Hello”;
   #endif /* PRODUCT_A */

   #ifdef PRODUCT_B
      char foo[] = “Beautiful”;
   #endif /* PRODUCT_B */

   #ifdef PRODUCT_C
      char foo[] = “Grey”;
   #endif /* PRODUCT_C */

   #ifdef PRODUCT_D
      char foo[] = “Goodbye”;
   #endif /* PRODUCT_D */

   char bar[] = ”World!”;
   printf(”%s %s\n”, foo, bar);

   return 0;
}
```

STRATEGIC REUSE

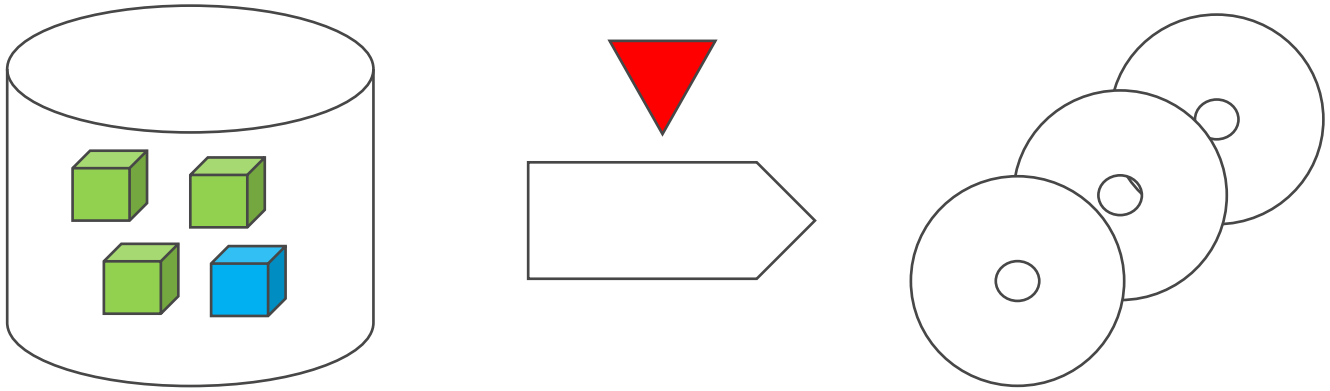
Explore

Variability & Commonality

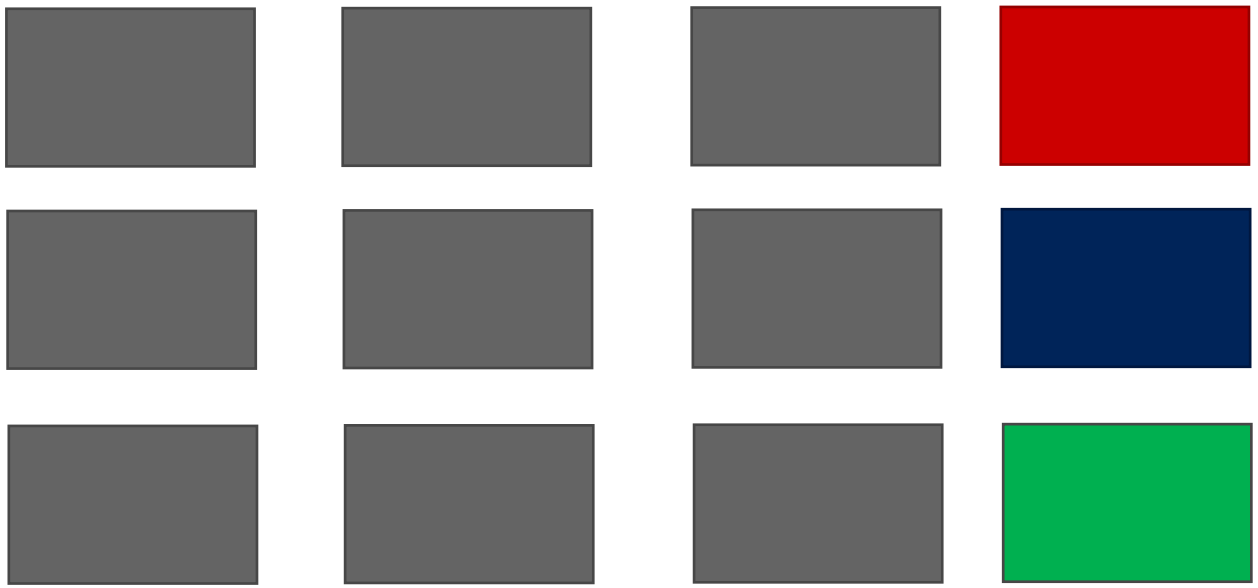
to achieve

Economy of Scale

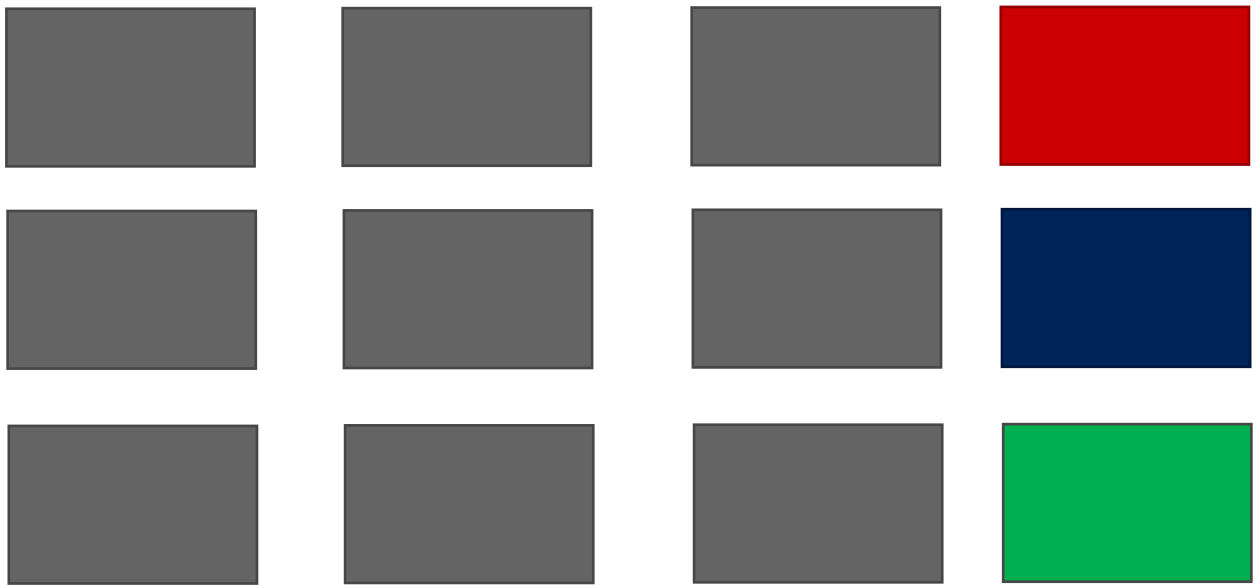
SOFTWARE PRODUCT LINES



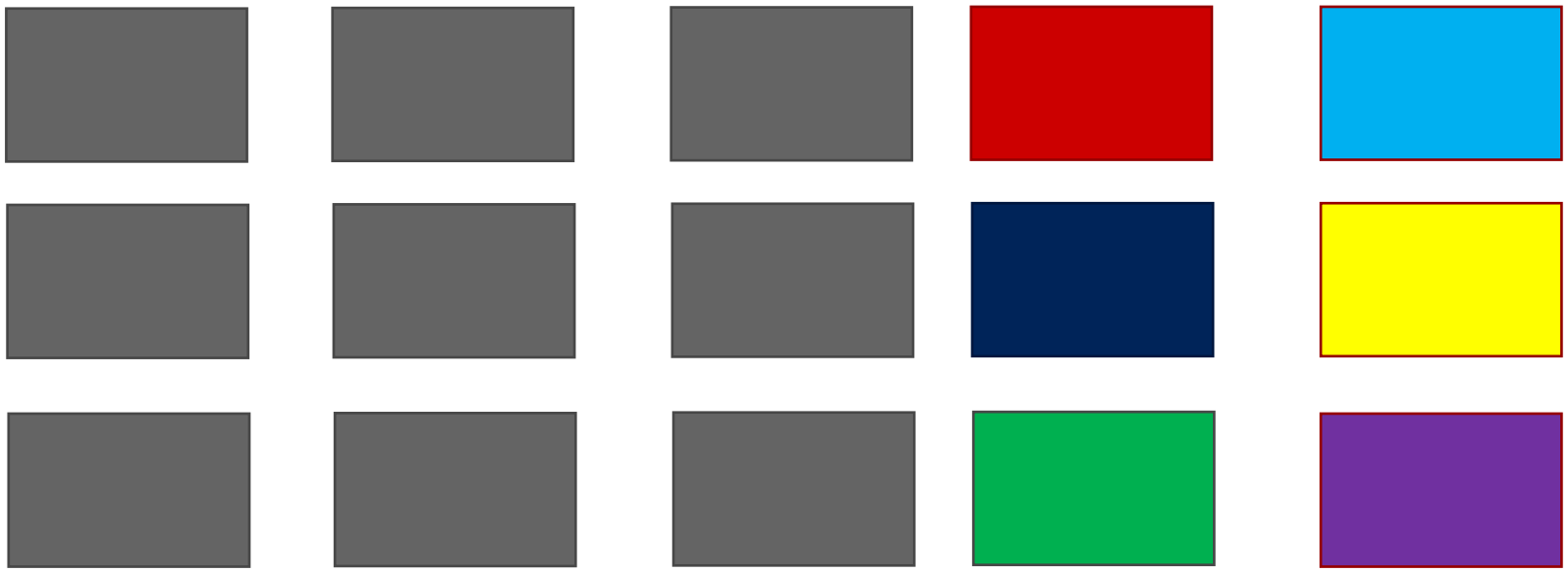
COMMONALITY & VARIABILITY



HOW MANY COMBINATIONS?



HOW MANY COMBINATIONS?



VERIFICATION HELL?

$$\binom{0}{0}$$

$$\binom{1}{0}\ \binom{1}{1}$$

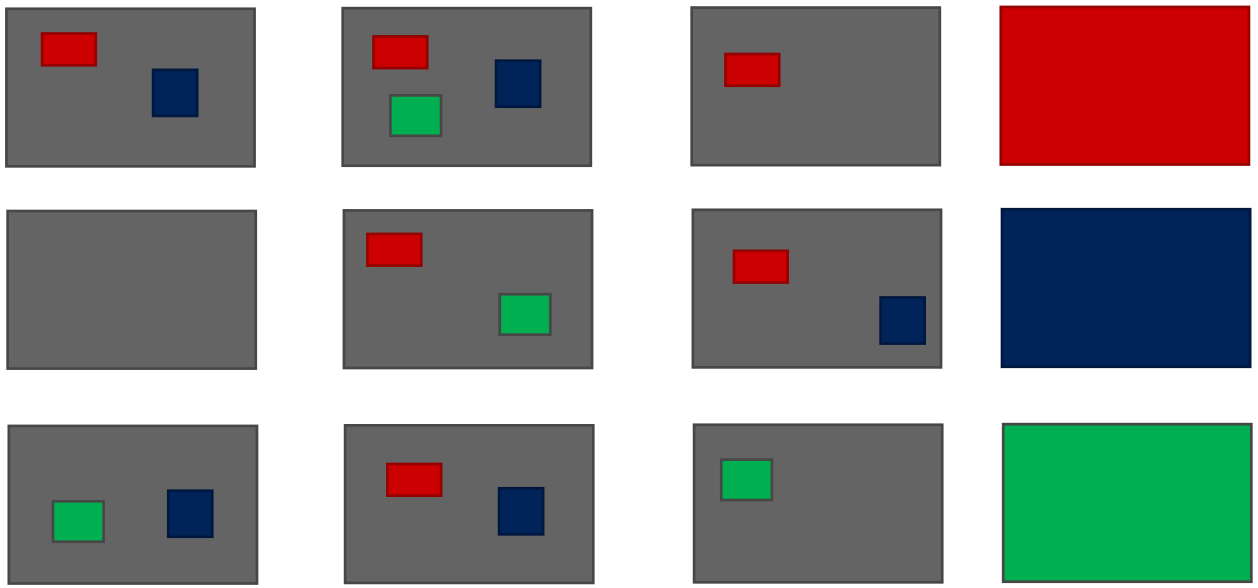
$$\binom{2}{0}\ \binom{2}{1}\ \binom{2}{2}$$

$$\binom{3}{0}\ \binom{3}{1}\ \binom{3}{2}\ \binom{3}{3}$$

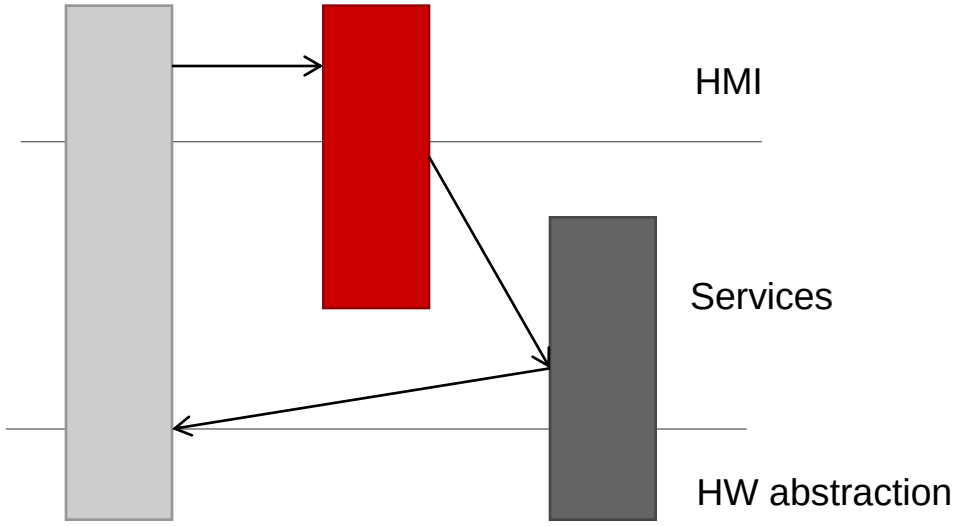
$$\binom{4}{0}\ \binom{4}{1}\ \binom{4}{2}\ \binom{4}{3}\ \binom{4}{4}$$

$$\binom{5}{0}\ \binom{5}{1}\ \binom{5}{2}\ \binom{5}{3}\ \binom{5}{4}\ \binom{5}{5}$$

VARIABILITY * COUPLING = COMPLEXITY

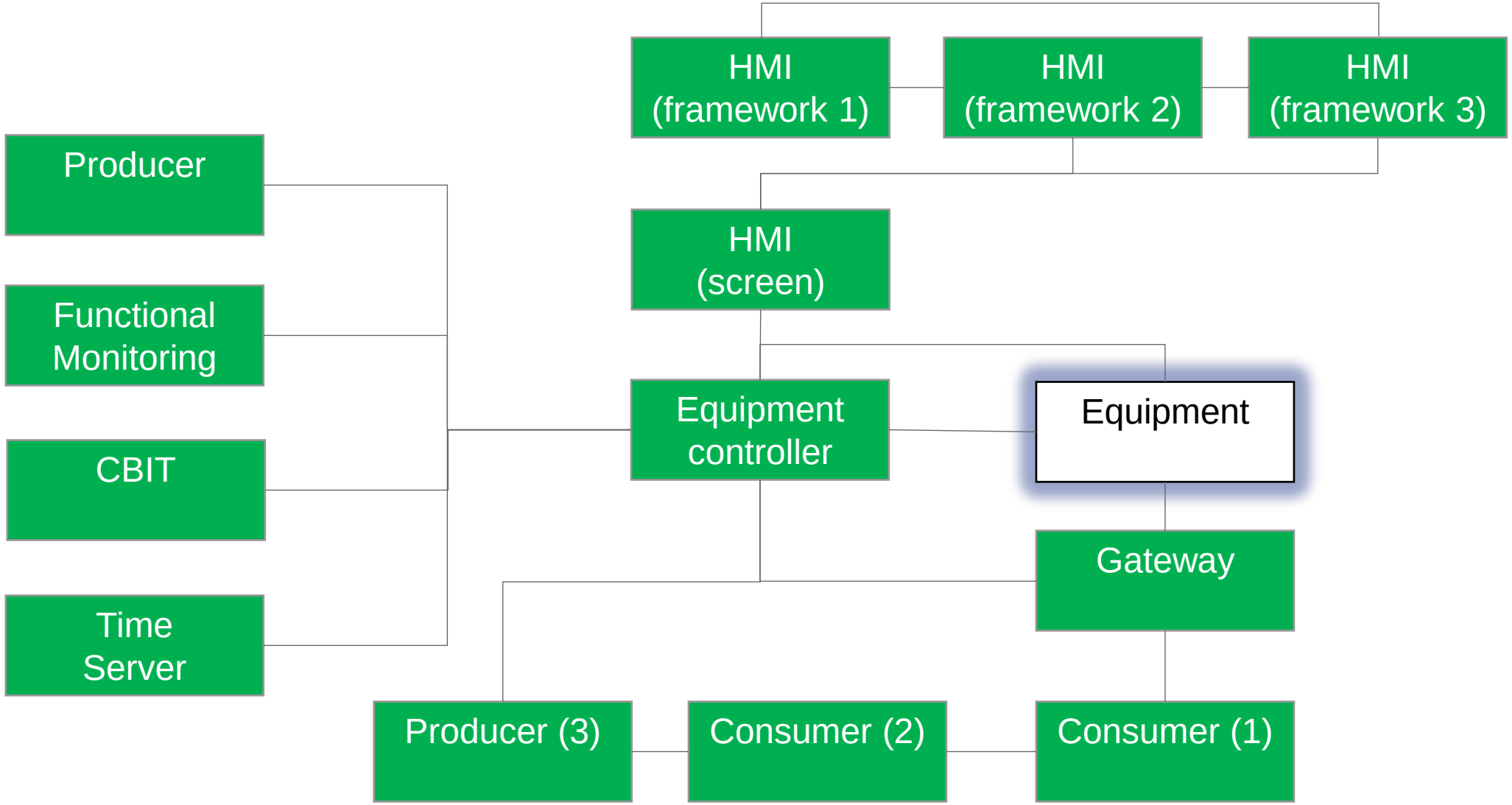


BIG BALL OF MUD

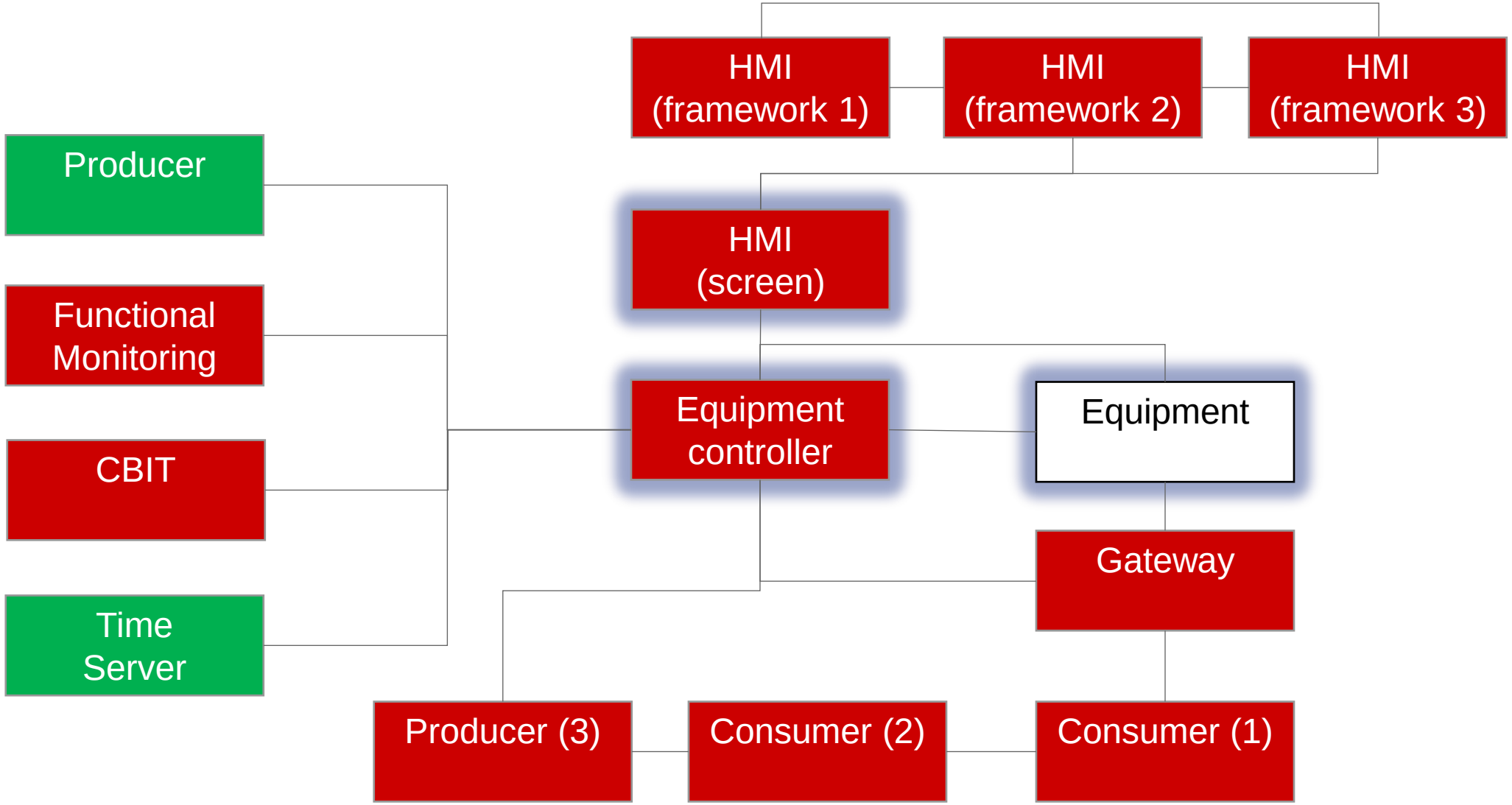


- Conway’s Law
  - "*organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations*“
- Leads to Cohesion principle
  - ”my module is where I put my code”

EXAMPLE

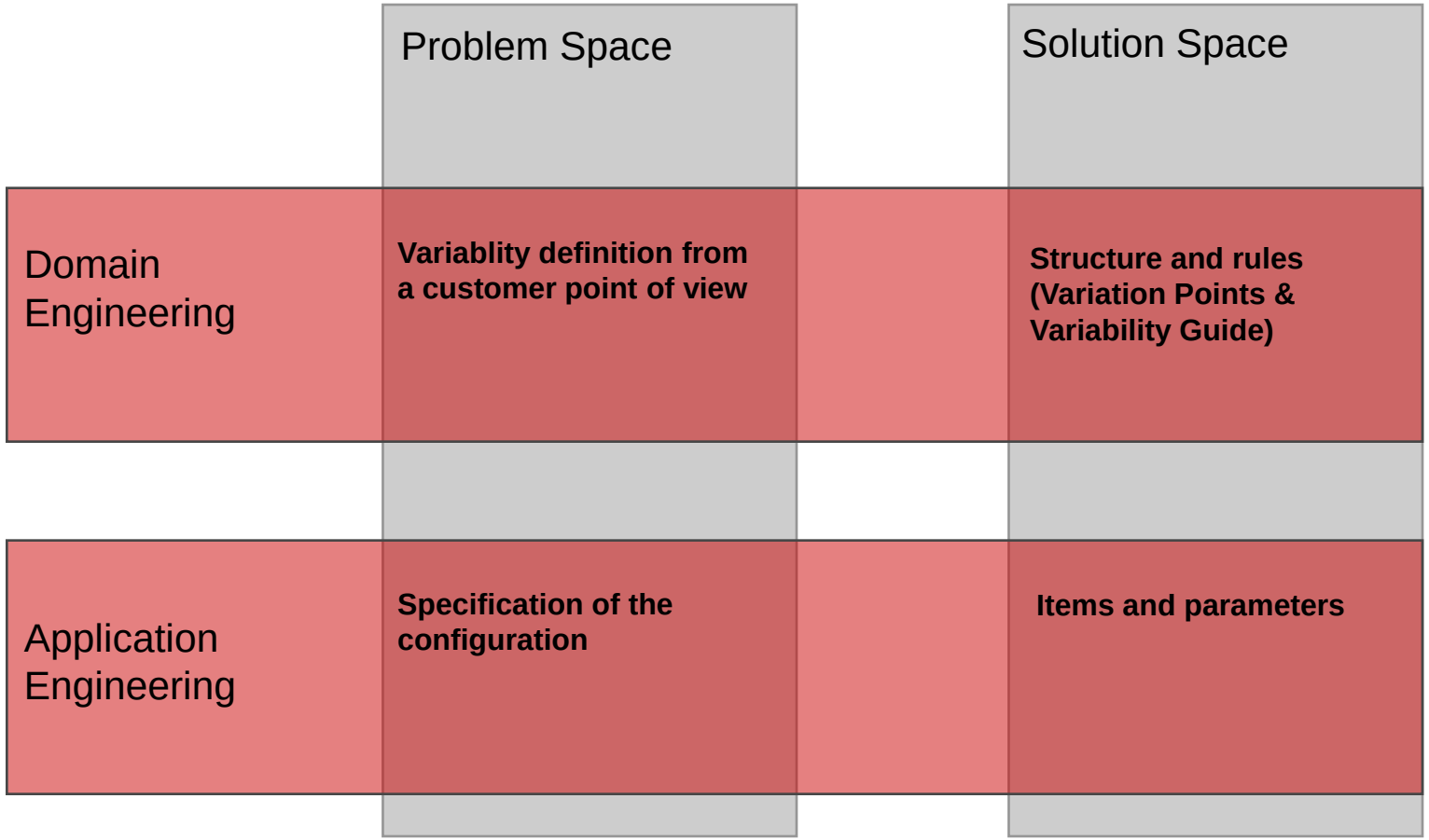

HMI (framework 1)
HMI (framework 2)
HMI (framework 3)
Producer
HMI (screen)
Functional Monitoring
Equipment controller
Equipment
CBIT
Gateway
Time Server
Producer (3)
Consumer (2)
Consumer (1)

EXAMPLE


HMI (framework 1)
HMI (framework 2)
HMI (framework 3)
Producer
HMI (screen)
Functional Monitoring
Equipment controller
Equipment
CBIT
Gateway
Time Server
Producer (3)
Consumer (2)
Consumer (1)

STATE OF THE INDUSTRY?

START WITH THE BUSINESS



ARCHITECTURE IS KEY

Tightly coupled system drives Integration- and Process centric Approach

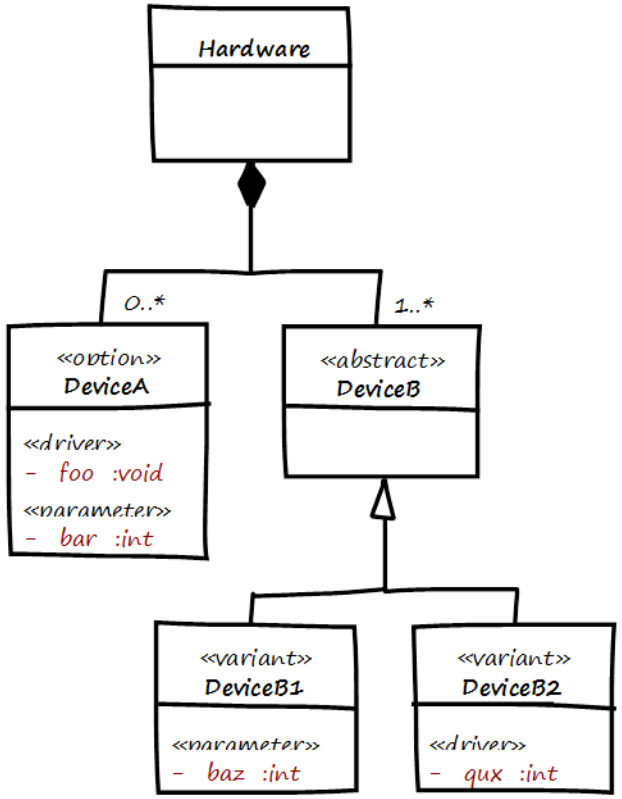
- –More or less centralized **synchronization of design**
- –Dealing with **ripple effects**

Loosely coupled system enables Composition and Architecture centric Approach

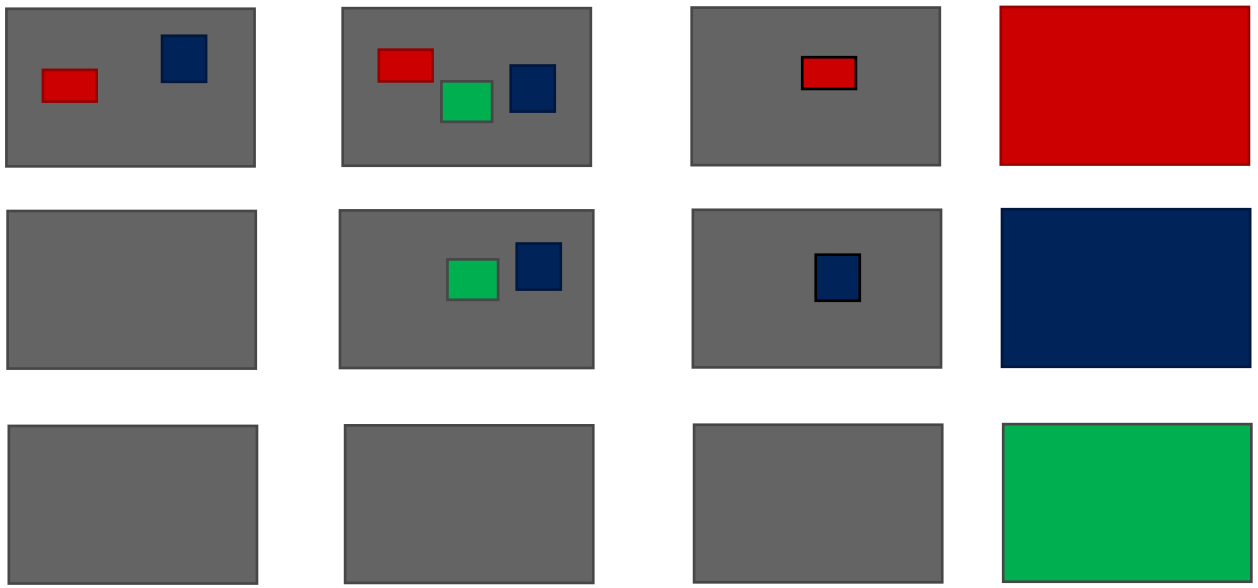
- –Architecture for **composition** and eco-systems
- –Backward **compability**

CORE PROBLEM – UNDERSTAND YOUR VARIABILITY

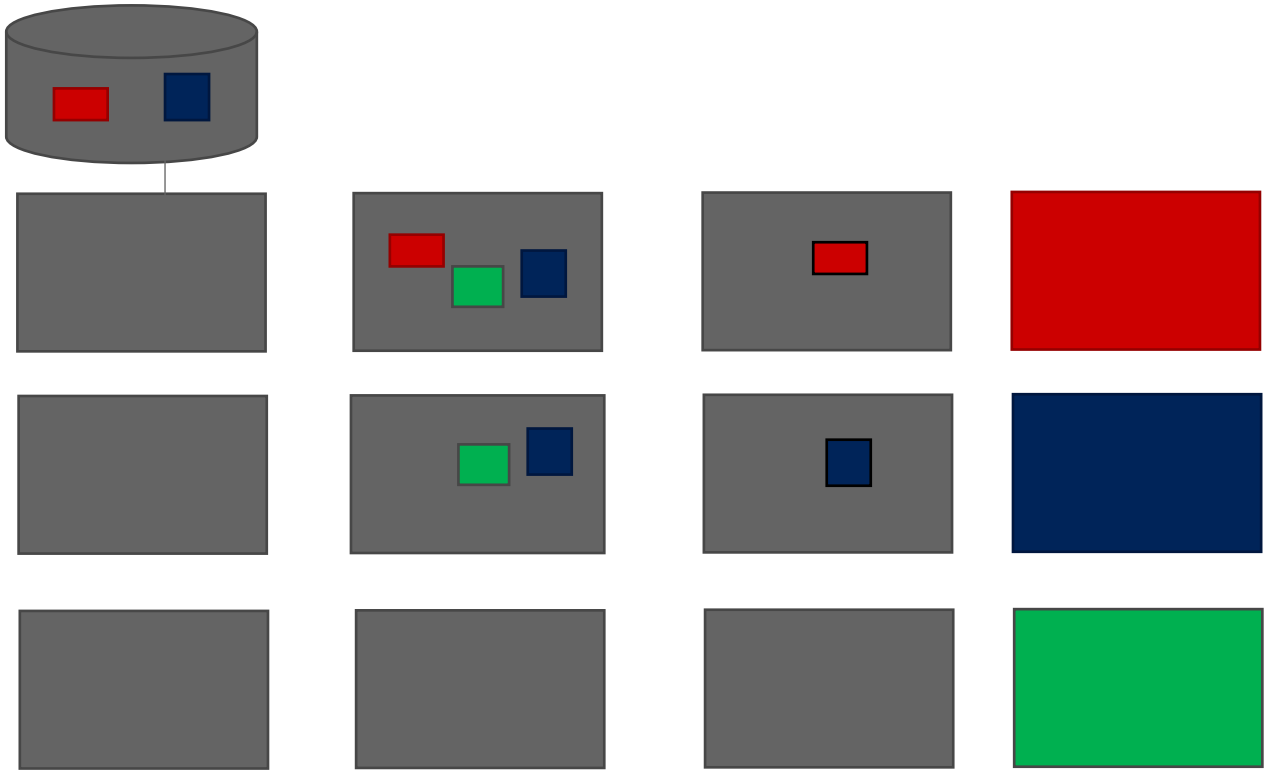
- Essential
  - Inherent in the problem domain
  - Differentiation is competitive advantage
- Accidental
  - Technical debt
  - Organization



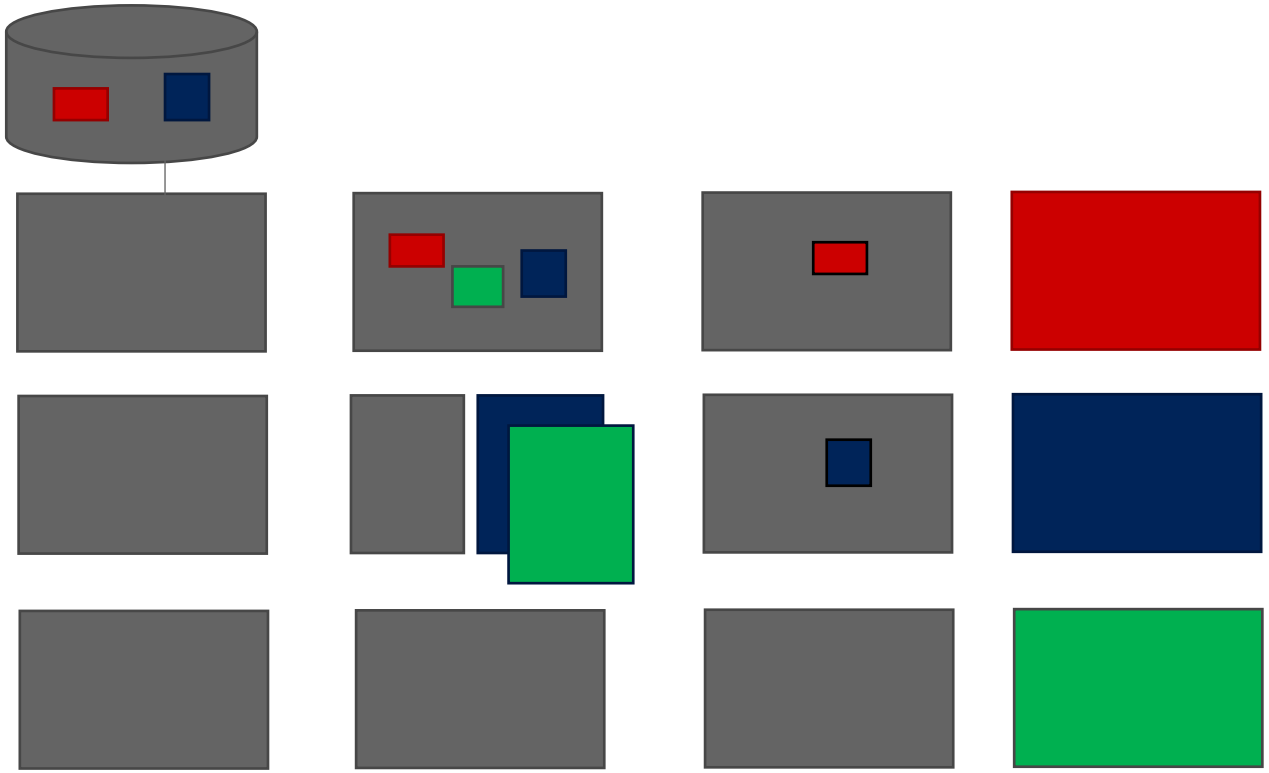
DECOUPLING STRATEGY



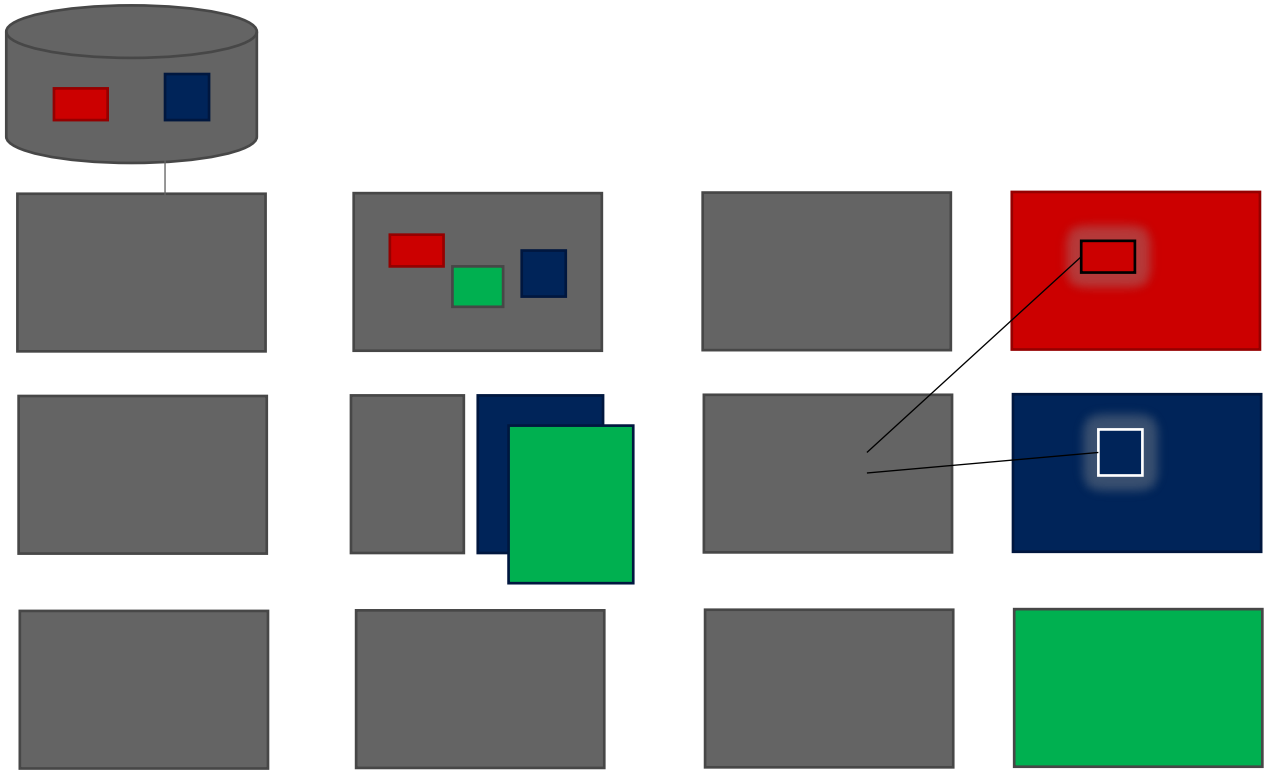
PATTERN – FACTORY (DDD)



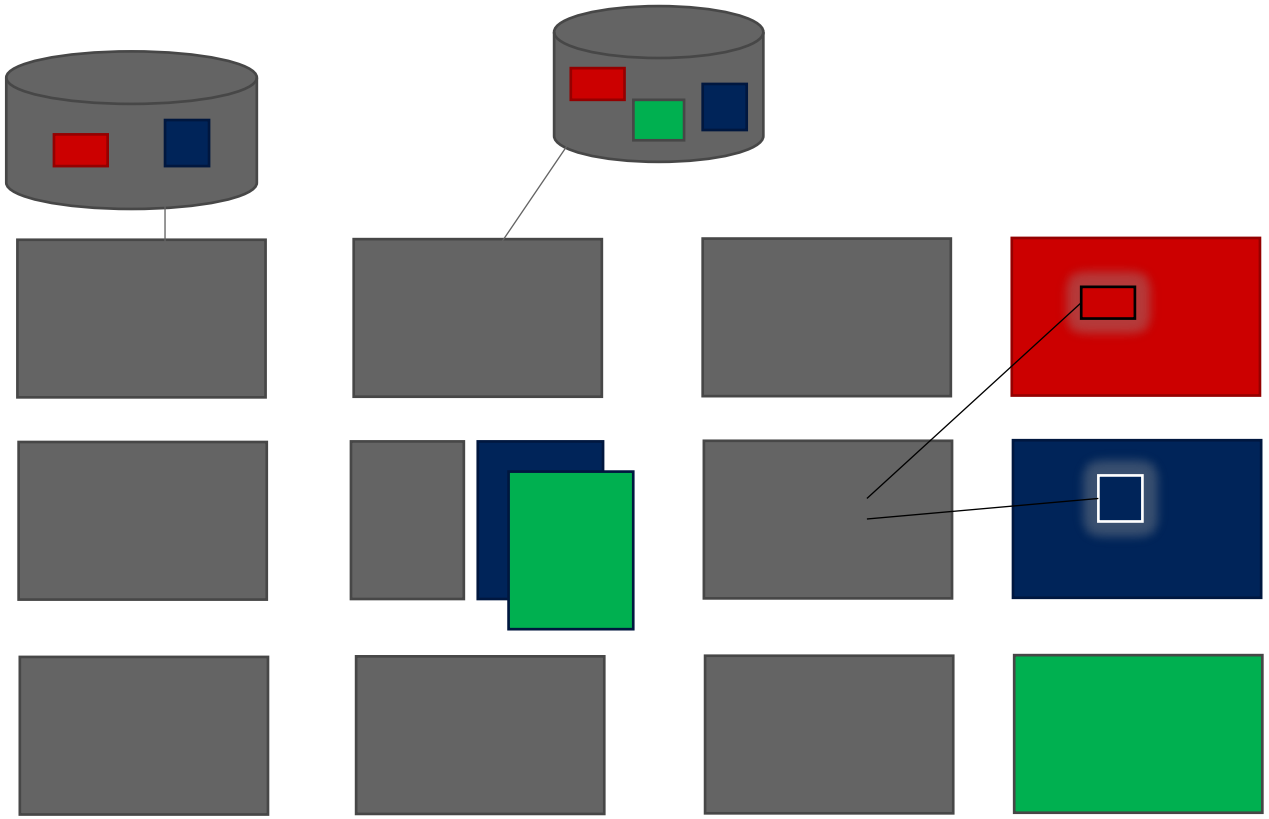
PATTERN – MODULE SPLIT



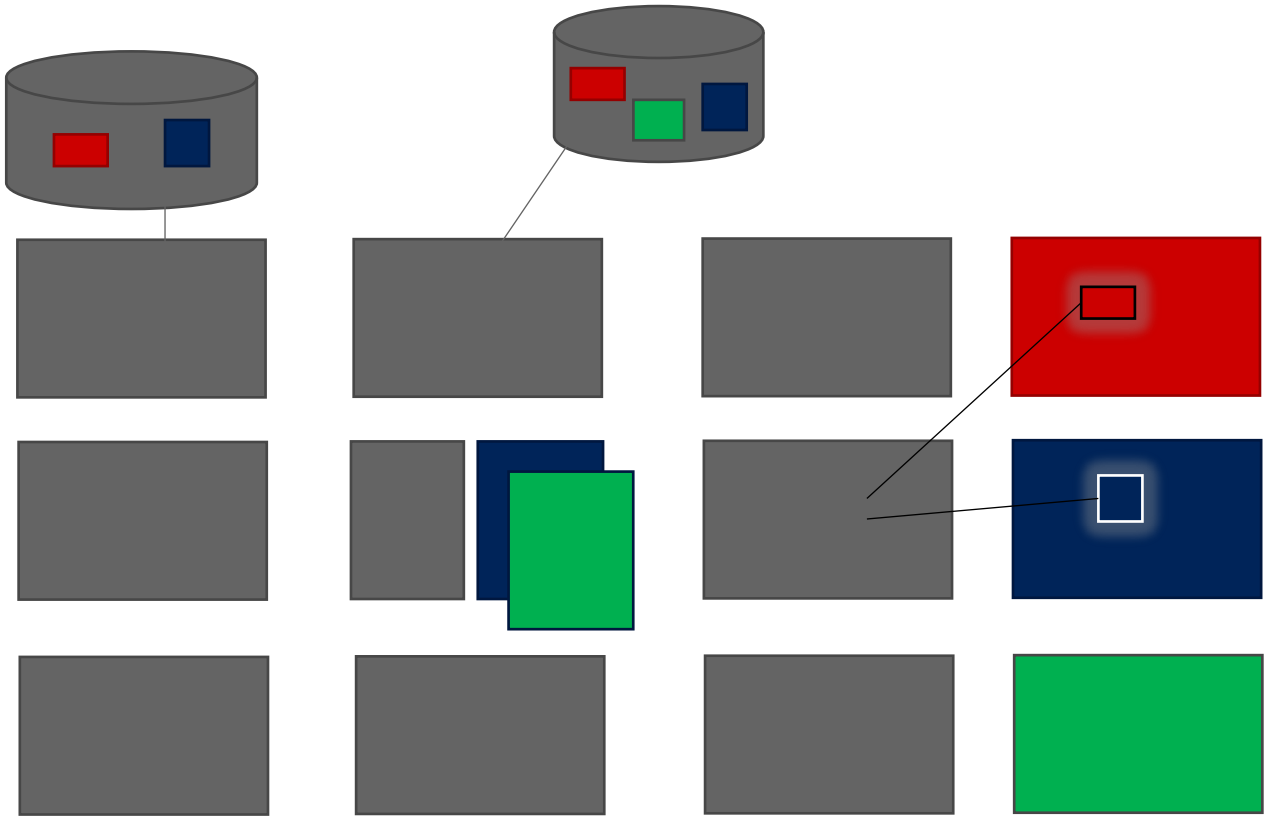
PATTERN – SELF CONTAINMENT



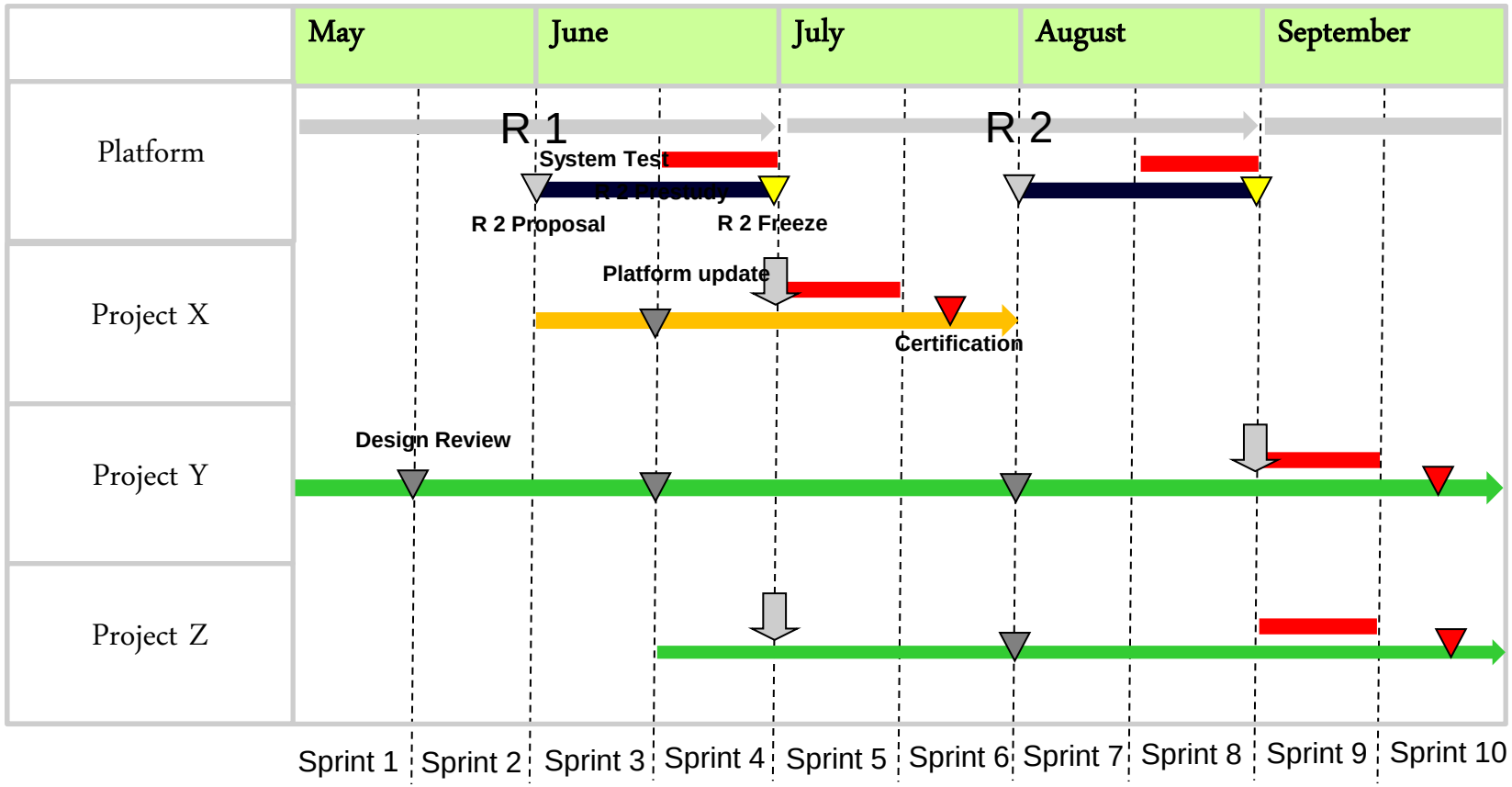
PATTERN – PARAMETERS



TREND: BUILD TIME -> RUN TIME



ORGANIZATIONAL DECOUPLING



QUESTIONS?


SAAB