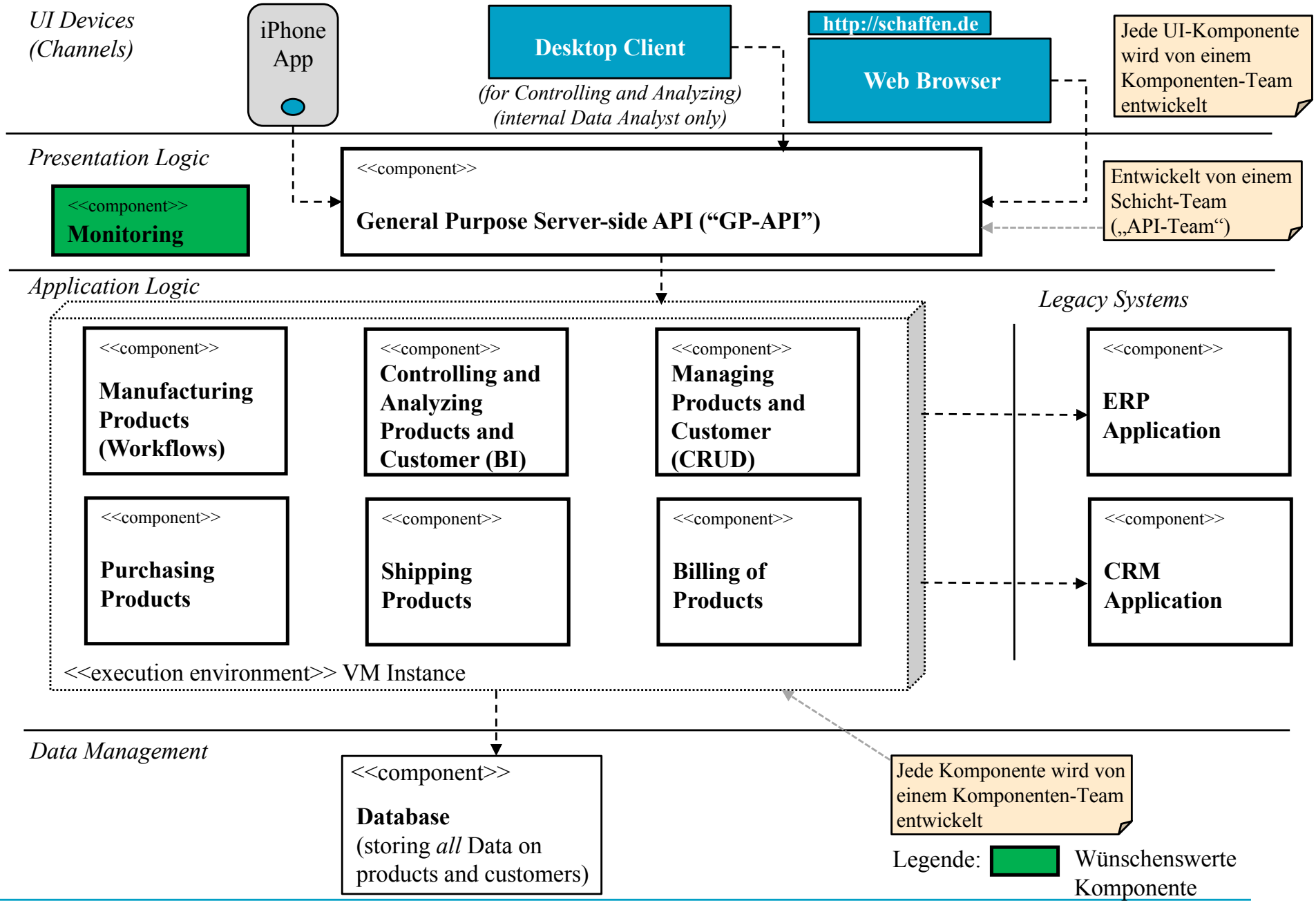
Hochschule
Bonn-Rhein-Sieg

Case Study Microservices – Analyse der Architektur der Firma WirSchiff*en*Das GmbH & Co KGaA

– internes Papier des Entwicklungsteams

Vertrauensvoll aufbereitet und bereitgestellt von:
Prof. Dr. Sascha Alda (c/o 2022)

Baustein-Sicht der Software-Architektur der Firma WirSchiff*en*Das GmbH & Co KGaA



Beschreibung der Produkte und Kunden-Akteure

- Entity „Product“:
  - Die Firma stellt Elemente (=Produkte) für Yachten her und verkauft diese für verschiedene Kundensegmente (siehe unten).
  - Verschiedene Produkte werden hier zusammengefasst:
    - Motoren für Yachten sowie
    - Einrichtungselemente für Yachten (beide *nur* über das Web-Interface bestellbar);
    - Kataloge über Yachten sowie
    - kleine Modelle von Yachten (beide *auch* über die Mobile App bestellbar)

- Akteure „Customer“:
  - Großkunden (kaufen vordefinierte Motoren und Einrichtungselemente über eine konventionelle Web-Anwendung, die in einem Web Browser abläuft)
  - Kleinkunden (benutzen ausschließlich eine Mobile App oder eine Web-Anwendung zum Erwerb von Katalogen und Modellen)
  - Premiumkunden (agieren wie Großkunden, können zusätzlich die Herstellung von Elementen *interaktiv* über eine Web-Anwendung mitgestalten)

Beschreibung der Komponente „Manufacturing Products“

- Diese Komponente wird von den Ingenieuren des Unternehmens zur Durchführung der Konstruktion (engl. Manufacturing) der Produkte für Groß- und Premiumkunden verwendet.

- In dieser Komponente gibt es für *jedes* Produkt (Motoren und Einrichtungselemente) für Yachten einen intern Workflow (Ablauf) zur Konstruktion der jeweiligen Produkte. Dieser interne Ablauf wird durch eine Konstruktions-Komponente realisiert. Eine Konstruktion eines Produkts kann innerhalb der Komponente *verwaltet* werden.

- Bei Motoren werden in dem Ablauf vor allem das „Optional Equipment“ abgefragt. Siehe dazu auch die Informationen aus dem Datenblatt für den Motor „Diesel Engine 2000 M96“.

- Für Premiumkunden gibt es eine interne Chat-Komponente innerhalb dieser Komponente, um Motoren interaktiv zu gestalten. Somit können Ingenieure und Premiumkunden online und gleichzeitig wichtige Entwurfsentscheidungen für das „Optional Equipment“ diskutieren. Eine Wiederverwendbarkeit der Chat-Komponente für andere Produkte für Großkunden ist gewünscht, bisher aber nicht realisiert worden.

Beschreibung der Komponente „Manufacturing Products“

- Es gibt ferner eine interne Analyse-Komponente, welche die aktuelle Spezifikation eines Produkts verifiziert. Die Laufzeit einer Analyse aufgrund interner Simulationsverfahren ist recht hoch, so dass es hier häufig zu Verzögerungen kommt, so dass die gesamte Komponente zur Durchführung der Konstruktion eines Produkts aufgrund der internen *engen* Kopplung häufig still steht, also *nicht* mehr *responsive* ist, bis die Analysekomponente sich wieder „erholt“ hat und eine Antwort liefert. Dies betrifft vor allem die Analyse von Motoren.

- Da die Analyse-Komponente von allen Workflows verwendet wird, kommt es gerade im Hochbetrieb der Anwendung (Oktober bis Dezember eines Jahres) zu sehr unangenehmen Verzögerungen und vereinzelt sogar zu Ausfällen im Ablauf. Eine *individuelle* Skalierung dieser Analysekomponente ist erwünscht durch die Ingenieure, wurde aber bisher durch die Bereichsleitung des Bereichs „Technik“ abgelehnt, da man die Skalierung der *gesamten* Anwendung vorziehen möchte. Eine individuelle Skalierung sei aus technischer Sicht schlichtweg *nicht* möglich.

- Auch die Ausfallsicherheit kann man nicht verbessern (O-Ton eines Insiders, der mit dem Bereichsleiter sprach: „*Damit sollen wir leben und halt nicht zu viel analysieren*!“). Dennoch ist man im Entwicklerteam der Komponente offen für die Anwendung von Pattern und Software-Bibliotheken zur Verbesserung der Ausfallsicherheit.

- Das Monitoring der gesamten Anwendung hält das Entwicklerteam für sinnvoll, gerade dann, wenn es zu einer Aufteilung nach Microservices kommt! Auch hier ist die Bereichsleitung sehr zögerlich. Man betrachtet eine Eigenentwicklung für das Monitoring als sinnvoll an (oder gibt es Produkte dazu?).

Beschreibung der Komponente „Manufacturing Products“ („Microservice“-Vision)

- Die in „Manufacturing Products“ angedeuteten Komponenten entsprechen Java-Packages; eine Komponenten-Orientierung hat man nicht realisiert bisher. Das Entwicklerteam wäre aber gerne an der Übernahme der Komponenten-Orientierung interessiert, gerne auch an einer zukünftigen Entwicklung der Komponenten als eigenständige Microservices.

- Konstruktionsspezifische Daten eines Produkts möchte man gerne „näher“ zu der Komponente „Manufacturing Products“ abspeichern, um somit nicht zu abhängig von der globalen Datenbank auf dem Layer „Data Management“ zu sein. Gegenüber einem universalen Datenmodell steht man sehr *kritisch* gegenüber.

- Man ist sich bewusst, dass durch das neue Daten-Management man nun auch Abhängigkeiten zu Komponenten bzw. zu den Entwicklerteams der Komponenten schafft, da fast alle auf (zumindest Teile der) konstruktionsspezifische Daten zugreifen. Das Team gibt sich offen und kundenorientiert! Man möchte hier gerne als „Zulieferer“ von Daten fungieren, sollten abhängige Teams *außerhalb* der Produktentwicklung hier Wünsche z.B. bei der Darstellung oder Aufbereitung von Daten haben. Innerhalb der Produktentwicklung will man partnerschaftlich arbeiten.

- Der Manager des Entwicklerteams gibt noch an, dass man mit dem „Controlling“-Team ein eher „angespanntes“ Verhältnis hat. O-Ton: „Eine „Zulieferung“ kommt hier nicht in Frage! Die sollen mal schön brav unsere Datenmodelle verwenden! Eine Abhängigkeit besteht zur (Teil-)Komponente „Managing Products“, da werden oft Daten von uns bezogen“. Hier gibt er sich sorglos: „Die müssen sich nicht anpassen oder was zuliefern, hier realisieren auf unserer Seite einen Adapter!“

Beschreibung der weiteren Komponenten

- Die Komponente „Purchasing Products“ ist eine Komponente für Kunden zum Kauf von Produkten.

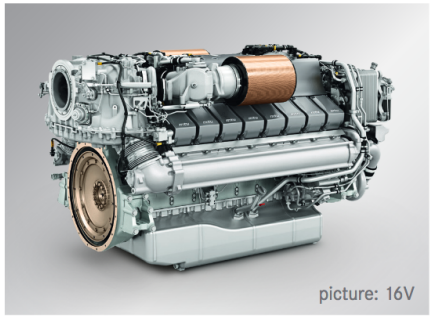
- Die Komponenten „Shipping Products“ und „Billing of Products“ sind interne Komponenten zum Versand bzw. zur Rechnungsstellung von Produkten. Diese werden durch Vertriebsarbeiter verwendet.

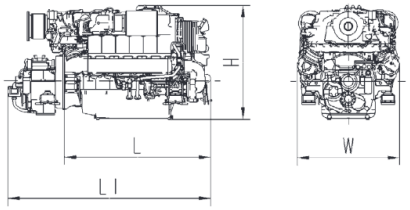
- Die Komponente „Managing Products and Customer (CRUD)“ ist eine interne Komponente um alle Produkte und Kunden intern zu verwalten. Sie wird von Vertriebsmitarbeiter, der Produktentwicklung und der Geschäftsführung verwendet. Diese Komponente verwendet die globale Datenbank auf dem Layer „Data Management“.
  - Das Entwickler-Team fühlt sich aktuell überfordert und fordert hier auf eine Aufsplitterung in zwei Komponenten bzw. auch in zwei Teams „Product“ bzw. „Customer“

- Die Geschäftsführung verwendet *auch* die Komponente „Controlling and Analyzing Products and Customer (BI)“.

Datenblatt für den Yacht-Motor „Diesel Engine 2000 M96“

Diesel Engines
10V/12V/16V 2000 M96





Dimensions and Masses

2000 M96
10V
12V [1]
12V [2]
16V

2000 M96 - with Gearbox type [3]
10V - ZF 2050
12V - ZF 2060 [1]
12V - ZF 2060 [2]
16V - ZF 3060

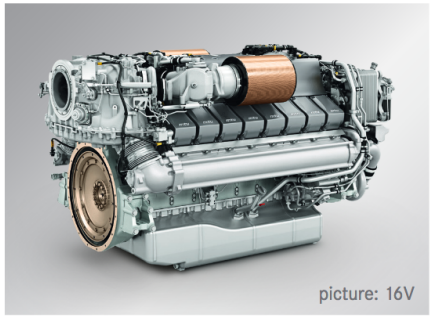
Optional Equipment	
Starting system	Air starter*
Auxiliary PTO	Alternator, 140A or 190A, 28V, 2 pole, bilgepump, on-engine PTOs
Oil system	Oil replenishment system, diverter valve for duplex filter
Fuel System	Duplex fuel pre-filter, diverter valve for fuel filter, monitoring fuel leakage
Cooling System	Coolant preheating system freestanding or engine mounted, integr. seawater gearbox piping
Exhaust System	90° Exhaust bellows discharge rotatable
Mounting System	Resilient mounts at driving end
Engine Management System	In compliance with Classification Society Regulations
Monitoring/Control System	BlueVision | NewGeneration
Power Transmission	Torsionally resilient coupling
Gearbox Options	Reverse reduction gearbox, el. actuated, gearbox mounts, trolling mode for dead-slow propulsion, free auxiliary PTO, hydraulic pump drives

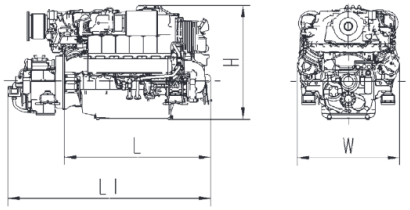
*only applies to 12V and 16V cylinder configurations

Quelle: Adaptiert von https://www.mtu-online.com/mtu/anwendungen/yacht-diesel-motoren/index.de.html

Bezug Analyse-Komponente zum Optional Equipment

Diesel Engines
10V/12V/16V 2000 M96





Dimensions and Masses

2000 M96
10V
12V [1]
12V [2]
16V

2000 M96 - with Gearbox type [3]
10V - ZF 2050
12V - ZF 2060 [1]
12V - ZF 2060 [2]
16V - ZF 3060

Optional Equipment	
Starting system	Air starter*
Auxiliary PTO	Alternator, 140A or 190A, 28V, 2 pole, bilgepump, on-engine PTOs
Oil system	Oil replenishment system, diverter valve for duplex filter
Fuel System	Duplex fuel pre-filter, diverter valve for fuel filter, monitoring fuel leakage
Cooling System	Coolant preheating system freestanding or engine mounted, integr. seawater gearbox piping
Exhaust System	90° Exhaust bellows discharge rotatable
Mounting System	Resilient mounts at driving end
Engine Management System	In compliance with Classification Society Regulations
Monitoring/Control System	BlueVision | NewGeneration
Power Transmission	Torsionally resilient coupling
Gearbox Options	Reverse reduction gearbox, el. actuated, gearbox mounts, trolling mode for dead-slow propulsion, free auxiliary PTO, hydraulic pump drives

*only applies to 12V and 16V cylinder configurations

O-Ton der zuständigen Ingenieurs, der im Rahmen einer ersten Anforderungsanalyse interviewt wurde:

„Die Analyse-Komponente für Komponenten ist deswegen so komplex, weil für jedes optionale Equipment ein eigenständiger und abgeschlossener Algorithmus zur Validierung und Simulation der Konfiguration durchgeführt werden muss. Aktuell sind diese Algorithmen in *einer* Klasse integriert. Ob man das nicht mal auftrennen könnte, indem man die Algorithmen in einer Choreographie *nebenläufig* abarbeitet? Die übrigen Komponenten sollte also *responsive* bleiben! Möchte auch gerne die aktuellen Status der einzelnen Algorithmen sehen können, das ist in der aktuellen Implementierung nicht möglich. Auch ein Retry eines einzelnen Algorithmus sollte möglich sein. Auch für eine Überwachung (Monitoring) der Services spreche ich mich aus.“

Quelle: Adaptiert von https://www.mtu-online.com/mtu/anwendungen/yacht-diesel-motoren/index.de.html