Vorüberlegungen

# Zusammenarbeit

1. Da wir jeden Mittwoch in den Treffen den Fortschritt präsentieren sollten wir uns evtl. Mittwoch Morgens-Mittags kurz treffen um den aktuellen Stand abzustimmen und zu klären was wir mit anderen Teams abstimmen müssen
2. Nach den Treffen am Mittwoch sollten wir uns zusammensetzen um zu planen welche Anforderungen wir in der nächsten Woche umsetzen wollen und wer davon was übernimmt
3. Montag vielleicht einmal Treffen um die Ergebnisse/Probleme der letzten Tage zu besprechen und zu planen dass wir bis Mittwoch alles vorgenommene auch geschafft bekommen
4. Sonstige Treffen und Slack-”Besprechungen” ansonsten einfach bei Bedarf

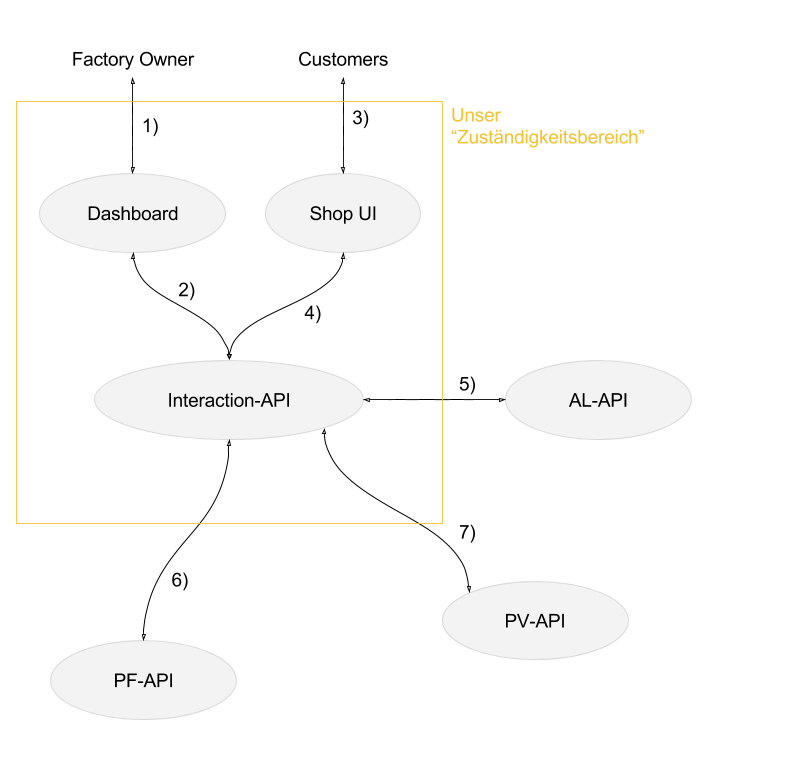
# Technologien

1. Eventuell Bitbucket zum Verwalten von Quellcode, Tasks, Bugs, etc., falls sich damit alle einigermaßen auskennen (Finde ich was schöner als das was der Lehrstuhl bereitstellt :-D)
2. Programmiersprachen und Frameworks hängen wohl davon ab was wir schon können und vor allem aber auch davon wieviel Wahlfreiheit wir überhaupt haben

# Architektur

Meine Idee wäre, dass jedes Team eine API bereitstellt über die es mit anderen Teams kommunizieren kann und hinter der Implementations Details verborgen sind. (Teams sind dann möglichst unabhängig voneinander)

Z.b. stellen wir die Interaction-API bereit, das Team “Automated Logistics” die AL-API, usw.



Annotationen:

1. Der Factory Owner kann über das Dashboard alle relevanten Informationen zu seinen Fabriken sehen und Aktionen wie starten und stoppen der Produktion ausführen
2. Die darzustellenden Informationen werden von der Interaction-API bezogen, bzw. werden ausgelöste Aktionen an diese weitergeleitet wo sie verarbeitet werden
3. Der Kunde kann über die Shop UI alle verfügbaren Produkte sehen, Bestellungen absetzen und seinen Bestellungsverlauf nachvollziehen
4. Die darzustellenden Informationen werden von der Interaction-API bezogen, bzw. werden Bestellungen an diese weitergeleitet wo sie verarbeitet werden
5. Wenn ein Kunde eine Bestellung aufgibt teilt die Interaction-API der AL-API mit wo die entsprechende Bestellung hin geliefert werden soll, andersrum kann die AL-API die Interaction-API nutzen um z.B. Änderungen im Status der Zustellung im Shop anzeigen zu lassen
6. Nach einer Bestellung benachrichtigt die Interaction-API die PF-API über eine neu eingetroffene Bestellung die ausgeführt werden muss, die PF-API nutzt die Interaction-API um Produktions- und Fabrik-Informationen dem Factory Owner und Kunden bereitzustellen
7. Nach einer Bestellung teilt die Interaction-API der PV-API mit welche Zutaten zur Herstellung an PF geliefert werden müssen, die PV-API kann der Interaction-API mitteilen welche Zutaten zur Verfügung stehen um diese im Shop anzuzeigen

# User Stories

## Customer:

1. As a customer, I want to fully customize and order goods via my own web application.
   1. Benutzereingaben werden über Shop UI erfasst und an Interaction-API weitergeleitet
   2. Interaction-API teilt Team PF-API mit was hergestellt werden soll
   3. (Interaction-API teilt Team PV-API mit welche Zutaten gebraucht werden) Bestellung der Zutaten wahrscheinlich eher Aufgabe des Teams PF
2. As a customer, I want goods I ordered to be delivered to my home.
   1. (Wie 1.a)
   2. Interaction-API leitet Lieferadresse an Team AL weiter
3. As a customer, I want that the delivered goods, which were ordered by me, conform to their order.
   1. Eher Aufgabe von Team PV/PF, da wir nur die spezifikation der Ware weiterleiten
4. As a customer, I want the web application to show me the live state of my order under production in the factory.
   1. Team PV, PF und AL können Interaction-API nutzen um den Status zu updaten (z.B. PF updated auf “Bestellung erhalten” -> PV auf “Zutaten geliefert” -> PF auf “Ware hergestellt” -> AL auf “In Auslieferung” -> ...)
   2. Shop UI bietet die Möglichkeit den Status anzuzeigen

## Factory Owner:

1. As a factory owner, I want that customers can order products via an ordering web application.
   1. (Wie 1.a)
2. As a factory owner, I want to offer the greatest diversity of products that can be produced by the factory production line capabilities.
   1. Team PF sollte wissen was alles hergestellt werden kann
   2. Team PF nutzt Interaction-API ums uns alle Möglichkeiten mitzuteilen
   3. Shop UI stellt alle Verschiedenen Produkte dar
3. As a factory owner, I want a product to be produced only if it was ordered by a customer.
   1. Hier sollte es nicht viel zu tun geben
4. As a factory owner, I want the factory to be aware of its capabilities
   1. Eher Aufgabe von Team PF/PV
5. As a factory owner, I want the machines to be easily exchangeable by similar versions without breaking changes of my factory’s software
   1. Eher Aufgabe von Team PF/PV
6. As a factory owner, I want to be able to get the detailed state of each machine at every point in time through its digital representation through a web dashboard application.
   1. Informationen werden durch API’s von Team PF/PV bereitgestellt
   2. Interaction-API nutzt andere API’s um Daten zu sammeln
   3. Dashboard nutzt Interaction-API um Daten darzustellen
7. As a factory owner, I want that agents supplied by platform providers optimize the production process using the digital representations of machines
   1. Eher Team PF/PV da wir nicht direkt an Produktion teilnehmen
8. As a factory owner, I want to be able to start and stop production from the web application.
   1. Dashboard kommuniziert mit API’s von Team PF/PV um Produktion kontrollieren
9. (Bonus: As factory owner, I want a 3D simulation of my factory to estimate the effects of automation machine changes.)
   1. 3D Simulation wäre unsere Aufgabe
   2. Auswirkungen abschätzen von Team PF/PV

## Agent Developer

1. As an agent developer, I want that there is a digital twin for each physical unit. I want that digital twins contain the entire business logic for their physical unit.
   1. Sollte uns (da keine Agents) nicht betreffen
2. As an agent developer, I want that each agent consists of a (digitally addressable) physical unit and its digital twin.
   1. Sollte uns (da keine Agents) nicht betreffen
3. As an agent developer, I want digital twins to be modeled as MontiArc architectural models.
   1. Sollte uns (da keine Agents) nicht betreffen
4. As an agent developer, I want physical units not to contain any business logic.
   1. Sollte uns (da keine Agents) nicht betreffen
5. As an agent developer, I want to be able to easily integrate the digital twin’s state into the factory web dashboard.
   1. Interaction-API kann benutzt werden um State im Dashboard zu hinterlegen
   2. Dashboard stellt Informationen dar
6. As an agent developer, I do not allow direct communication between physical units. Communication must be realized via their digital twins.
   1. Sollte uns (da keine Agents) nicht betreffen
7. As an agent developer, I want the communication between agents to conform to the AutomationML standard.
   1. Sollte uns (da keine Agents) nicht betreffen
8. As an agent developer, I want that each physical unit has a REST interface.
   1. Sollte uns (da keine Agents) nicht betreffen
9. As an agent developer, I want that each digital twin communicates with its physical unit via the REST interface of the physical unit.
   1. Sollte uns (da keine Agents) nicht betreffen
10. As an agent developer, I want to be able to reflect the cost of machine actions in the agent’s interface.
    1. Wir stellen kosten in Dashboard dar

## System Designer

1. As a system designer, I want that agents know how to communicate with each other. All communication should follow a common communication protocol.
   1. Sollte uns (da keine Agents) nicht betreffen
2. As a system designer, I want that an agent does not know any details about any other agent. Agents only know how to communicate.
   1. Sollte uns (da keine Agents) nicht betreffen
3. As a system designer, I do not want any agent to know the production plan of any product.
   1. Sollte uns (da keine Agents) nicht betreffen
   2. ?
4. As a system designer, I want each product to be produced to know its production plan and that the production plan is communicated to the agents together with an order.
   1. Sollte uns (da keine Agents) nicht betreffen
5. As a system designer, I want a dispatcher to provide the ordering web app.
   1. Store UI wird von uns bereitgestellt
6. As a system designer, I want the dispatcher to maintain a queue with the products to be produced.
   1. Gemeinsame Aufgabe von Team PF/PV und uns
7. As a system designer, I want that the dispatcher creates a production plan for each product to be produced.
   1. Gemeinsame Aufgabe von Team PF/PV und uns
8. As a system designer, I do not want the production plan to contain concrete agents for specific tasks.
   1. Gemeinsame Aufgabe von Team PF/PV und uns

# Offene Fragen

1. Welche Freiheiten haben wir in der Wahl, wie wir die Anforderungen umsetzen?
2. Wer ist für die Dispatcher verantwortlich? Eigenständiges System? Jedes Team verwirklicht seine Schnittstellen?