Microservice-Architektur für eine Verteidigungsanlage

In einer Vertigungsanlage erfolgt ein komplexer Produktionsablauf. Die folgenden Überlegungen basieren auf einem hypothetischen Konzept für eine Passpersonalisierungsmaschine.

Aufgabe der Maschine

Die Aufgabe einer Passpersonalisierungsmaschine, nachfolgend als "Personalisierungsmaschine" bezeichnet, besteht darin, ein vorhandenes Passdokument weiter zu personalisieren. Ein Dokument (Pass) wird im geschlossenen Zustand auf der einen Seite in die Maschine eingeführt und soll auf der anderen Seite vollständig personalisiert in einem geschlossenen Zustand ausgegeben werden. Die Ein- und Ausgabe erfolgt in vertikalen Containern.

Aufbau der Maschine

Die Maschine soll aus mehreren Stationen bestehen, welche unterschiedliche aufgaben erfüllen. Für unsere betrachtung sind alle Actoren und Sensoren der Maschine relvant. Die Actoren können durch die Softwar zu einer Bewegung gebracht werden, wärend Snsoren werte an die Software Liefern können. Eine Baugruppe kann sowohl aus einem oder mehreren Actor als auch aus einem oder mehreren Sensor bestehen. Sie kann auch keinen Sensor oder Actor beinhalten. Es gibt auch Baugruppen ohne Sensor und Actor. Dies sind für die Betrachtung uninteresannt.
Grundsetzlich muss die Bewegung der Maschine gesteuert werden. Typischerweise wird hierfür eine oder in Komplexen maschinen auch mehrer SPS'n verwendet um die Werte der Sensoren auszulehsen und die Actoren entsprechend zu bewegen. Es gibt demnach eine Zentralle rechenkomponente und einen Monolitischen aufbau des Programmes.
Um eine steuerung für eine solche Maschine in form von Microserfices umzusetzen muss dieses Hauptprogramm in mehrere Bestandteile aufgeteilt werden. Dabei können folgende Schritte genommen werden:

1. Jede **Station** erhält ein eigenens Programm. Dieses gibt die zu befolgenden Abrebeitschritte in dieser station an. Da jede station ein eigenes Programm erhält wäre es möglich dies auf dedizierten chips für jede Station einzelnd auszuführen.
2. Jede **Baugruppe** in einer station erhält eine eigenen programierung. Eine Baugruppe ist grundsetzlich ein Zusammenschluss von mehreren bestandteilen der Maschine. In gewisser weise wird jede Station in weitere Substationen unterteilt. Diese erhalten ihre eigenen Programierung und im bestenfalle auch eigene Rechenchips.
3. Jeder **Actor** und **Sensor** erhält eine eigenes Programm und dedizierte Hardwäre. Dies ist nicht dringend untypisch. Dabei besitzen insbesondere größere motoren rechtheufig dedizierte Hardare, auf welchen dedizierte Software zur steuerung läuft. Es ist aber untypisch dies für jeden einzelnen senso und Zylinder umzusetzen.

Betrachtung der Ansätze

Im grundkonzept ohne Microserfic wird spezialisierte Hardware und software zur steurung verwendet. Diese hardawre ist vergleichsweise teuer aber für diese anwendung gedacht.

Microservic auf Ebene von Stationen

Die umsetzung der Steuerung auf stationsebene ist grundsetzlich nicht unüblich und wird50 insbesondere dann verwendet, wenn die Komplexität der einzelnen Stationen recht hoch ist. In diesem falle erscheint die aufteilung des REchenaufwandes sinnvoll. Eine einzelne SPS könnte die gesamte Maschine warsccheinlich schlecht steuern. Gleichzeitig wird hierbei der Sinn eines Microservices nicht erreicht, da jede Station so komplex ist, dass sie eine eigens Monolitysche software benötigt.
Im falle unserer Passmaschine sind einzelne Stationen eigentlich nicht so komplex, dass diese nicht von einem Zentralen gerät gestuert werden können. Die aufteilung der Rechenarbeit auf die Einezlen Stationen ist demnach nicht nötig und es findet eine vereinfachung der Programmteile stat.
Es wäre sinnvoll eine Aufteilung des Programmes in einzelne Module für die einzelnen Stationen umzusetzen. Dies solten aber immer noch auf einem Gerät ausgeführt werden. Dies liegt hauptsechlich daran, dass die Programme der einzelne Stationen immernoch besondere Harxware benötigen und es aus Ekonomischer sich sinnvolle isteine SPS gut auszulasten als viele wenig ausgelastet SPS zu haben. Außerdem geschtaltet sich die Komunikation zwischen den Stationen einfacher. Dies ist insbesnodere für Echtzeit kreitereien wichtig, auf welche später weiter eigegeange werden soll.
Es ist demanch sinnvoll eine Microserficartige softwararchitectur auf ebene von Stationen für eine solche Maschine umzusetzen.

Microservic auf Ebene von Baugruppen

Soll auf eben von Baugrun eine Microservic Architectur umgesetzt werden muss folgendes beachtet werden:

1. Jede **Baugruppe** solte ein eigense verarbeitungs gerät erhalten. Es wird darauf gehofft, dass die Verarbeitung der Aufgaben einzelner Baugruppen so einfach ist, das hierfür nur wenig rechenkapazität benötigt wird.
2. Die **Baugruppen** müssen miteinander verbunden werden.
3. Ein komunikationsstandart zwischen den Baugruppen muss entwickelt werden

Aufgabe eines Baugruppenrechners ist die Verarbeitung aller singale der Sensoren, welche zur Baugruppe gehören, sowie die Steuerung aller Actoren der Baugruppe. Dabei muss jede Baugruppe anforderungen von anderen Baugruppen befolgen, welche eine Bewegung von der Baugruppe fordern, sowie die verarbeiteten signale der Sensoren weitergeben, wenn diese von anderen Baugruppen benötigt werden.
Ein Probelem der Abklärung von zuständigkeiten entsteht. Wer darf wem sagen, was dieser tuhen soll.
Ein weiteres Probelm bei diesem aufbau ist die Komunikationsart. Eine kabellose vunkverbindung wäre vorteilhaft, da hiermit viele kabel gespart werden könnten. Doch eine umsetzung über Bsp. WLAN scheint unter der beachtung von echtzeitkritereien äuserst unrealistisch. Daher müssten alle Baugruppen miteinander verbunden werden, welches zu einem erheblichen kabel wirwar führen würde. Dies Problematik besteht auch in der nachfolgenden Umsetzung.

Microservic auf Ebene von Bauteilen

Wird die Arbeit nochw eiter verteilt gelangt man zur letzmöglichen stufe. In dieser besitzt jeder Aktor und Sensor eine eigenen verarbeitungs Logik und warscheinlich auch einen eigenen chip. Jedes glied hat nur eine Minimale aufgabe umzusetzen. Sensoren müssen daten interpretieren und an die entsprechenden Aktoren weiterleiten. Actoren müssen auf Inforamtionen der Sensoren reagieren und womöglich selber einen Arbeitszusstand weiterleiten. Auch das verkabelungs problem könnte gelöst werden. Grundsetzlich müssen die sensoren sowieso eine Datenleitung besitzen, damit diese informationen Weitergeben können. Diese könnten zur verteilung von Inforamtionen verwendet werden können.

Überspannende Probleme

Die umsetzung eine Microservizes auf ebene von Stationen erscheint sinnvoll, sovern diese auf möglichst wenigen unterschiedlichen geräten ausgeführt wird. Bei allen klein teiligeren aufteilungen entstehen folgenden Probleme:

1. **Komunikation**:
    1. Es ist unklar, wer welche **zuständigkeiten** hat
    2. **Race conditions** können vermehrt auftreten. Grade die Unsicherheit von Netzwerken können viele Racekonditions entstehen lassen.
2. Umsetzung von **Echtzeitkriterien**: Die umsetzung von echtzeitkriterien erscheint schwer. Netzwerk komunikation ist heufig nicht sicher und kann eine unbekante Zeit dauern. Die entwicklung, betrachtung und umsetzung von echtzeitanforderungen erscheint schwer.
3. **Sicherheit**: Insbesonder in Europa müssen gewise sicherheits standarts eingehalten werden. Heufig wird hierfür spezialisierte Hardware benötigt. Eine verteilung der Aufgabe ist potenziell unsicher oder auch kostspielig
4. **Entwicklung**: Das Korekte zusammenspiel von Hunderten Sensoren und Actoren zu koordinieren ist bereits in einer Monolitischen software schwer. Wird diese aufgeteilt wird erscheint der aufwand nicht geringer zu werden. Die änderung an einer Baugruppe kann eine Softwareänderung an vielen anderen stellen zur folge haben. Dabei muss zuerst ersichtlich werden, welche dier vielen Services verändert werden müssen und dann müssen dies auch tatsechlich alle verändert werden.
5. **Übersichtlichkeit**: Die übersichtlichkeit des Ablaufes wird verschlechtert. Es gibt keine eindeutigen ablaufketten diese werden über datenaustauschen zwischen einzelnen Microserfices umgesetzt. Dieser austausch muss angelegt und verstanden werden. Warscheinlich wäre ein eigenes Visualiesierungstool nötig.

Zusammenfassung

Zusamngefast scheint die Umsetzung einer Passmaschien in einer kleineren Microservic architektur als auf basis von Stationen nicht sinnvoll. Die entwicklung dieser ist erhblich schwerer und unübersichtlicher. Auch Profitiert ein solche mashcine nicht von den vorteilen eines Microservices. Es wird zu jedem Zeitpunkt genau eine verarbeitungs Instanz pro aufgabe benötigt. Der rechen aufwand sollte relativ konstant sein.
Die umsetzung in form von einem Microservic Pro station kann empfohlen werden doch eine weiter aufspaltung ist nicht sinnvoll.