Qualifikationsverfahren Informatikerin/Informatiker EFZ

Individuelle praktische Arbeit (IPA)
Aufgabenstellung

Form: 01

Kandidatin/Kandidat		
Nummer:	Name: **KESSLER**	Vorname: **Remo**
Lehrberuf: **Applikationsentwicklung**		

Schultage:	Mo	Di	Mi	Do	Fr	Sa
Vormittag:	☐	☐	☒	☐	☐	☐
Nachmittag:	☐	☐	☒	☐	☐	☐

Lehrbetrieb bzw. Praktikumsbetrieb	
Name des Betriebs: **CM Informatik AG**	
Strasse: **Sirnacherstrasse 7**	
PLZ: **9500**	Ort: **Wil**

Vorgesetzte Fachkraft	
Name: **Tinner**	Vorname: **Martin**
E-Mail: **martin.tinner@cmiag.ch**	Telefon direkt: +41 71 421 34 14

Fachexperten		
Experte 1 (Hauptexperte)	Name: **Schättin**	Vorname: **Patrick**
Experte 2	Name: **Loser**	Vorname: Reto

Kurztitel der Arbeit
Kurztitel (max. 80 Zeichen): **Zentralisierte Parameterverwaltung für eine Mikroservices-Architektur**

Zeitverhältnisse				
Starttermin: 19.03.2018	Endtermin: 06.04.2018	Vorgabezeit [h]: 80	Anteil Praxis [%]: 50	Anteil Dokum. [%]: 50

Datum und Unterschrift der vorgesetzten Fachkraft
Mit der Unterschrift bestätigt die vorgesetzte Fachkraft, dass die notwendigen Voraussetzungen im Betrieb bezüglich Arbeitsplatz, Material und Betreuung vorhanden sind, dass die Fachkraft die Aufgabenstellung selber verfasst hat und dass sie die Arbeit begleiten und die Dokumentation kompetent korrigieren wird:

Datum und Unterschrift der Kandidatin/des Kandidaten
Mit der Unterschrift bestätigt die Kandidatin/der Kandidat, dass er mit dieser Aufgabenstellung einverstanden ist:

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)

Ziele, die zur Erfüllung der Arbeit zu erreichen sind (z.B. nach SMART):
Ziel 1: **Benutzerfreundliche, zentrale Verwaltung**
Ziel 2: **Selbstdokumentation und Validierung**
Ziel 3: **Rücksicht auf MicroServices Architektur**

Für den Lernenden in dieser Arbeit neue Themen (Summe <= 20%!)		
Thema	**Ant.[%]**	**Fachliche Betreuung sichergestellt durch:**
Thema 1: **MicroServices Architektur**	Anteil: **10**	Name und Vorname der Fachperson: Matthias Hess
Thema 2: **RabbitMQ / MassTransit EventBus**	Anteil: **10**	Name und Vorname der Fachperson: Matthias Hess

Spezielle Anforderungen an das Umfeld, Infrastruktur, Budget, Bemerkungen:

- Es dürfen keine schützenswerte Daten in Screenshots und an der Präsentation gezeigt werden.
- Die Inhalte dieser IPA dürfen nicht weitergegeben werden.
- Wichtig ist der Hinweis, dass es sich um ein laufendes Projekt des BAR handelt

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)

Für die Bewertung der Arbeit spezifische zusätzliche Fachkriterien (7 Kriterien)		
Kriterium 1: **Benutzerfreundliche, zentrale Verwaltung**	Anforderungen: **- Der Administrator kann alle Parameter aller Systemdienste an einem Ort pflegen** **- Das Auffinden des gewünschten Parameters wird vom System unterstützt** **- Defaultwerte sind optisch als solche identifizierbar** **- Validierungsfehler werden opisch hervorgehoben**	Gütestufen: 3: **vier Anforderungen erfüllt** 2: **drei Anforderungen erfüllt** 1: **zwei Anforderungen erfüllt** 0: **eine oder keine Anforderung erfüllt**
Kriterium 2: **Selbstdokumentation und Validierung**	Anforderungen: **- Ein Hilfetext kann für jeden Parameter hinterlegt werden** **- Die Validierung erlaubt die Kontrolle bei der Erfassung** **- Die Validierung kann auch separat aufgerufen werden (bei der Installation)** **- Der Validierungsmechanismus soll versch. Formate unterstützen (vgl. Aufgabe).**	Gütestufen: 3: **vier Anforderungen erfüllt** 2: **drei Anforderungen erfüllt** 1: **zwei Anforderungen erfüllt** 0: **eine oder keine Anforderung erfüllt**
Kriterium 3: **Rücksicht auf MicroServices Architektur**	Anforderungen: **- Die Parameter werden im Service gespeichert, nicht zentral(!)** **- Die Services funktionieren auch, wenn die Parameterverwaltung (PV) nicht läuft** **- Die PV funktioniert rechnerübergreifend, ohne FileSharing** **- Die PV erkennt selbstständig die parametrierbaren Dienste und ihre Werte**	Gütestufen: 3: **vier Anforderungen erfüllt** 2: **drei Anforderungen erfüllt** 1: **zwei Anforderungen erfüllt** 0: **eine oder keine Anforderung erfüllt**
Kriterium 4: **Programmseitig einfach einsetzbar**	Anforderungen: **- Es werden sprechende Namen für Klassen / Methoden verwendet**	Gütestufen: 3: **drei Anforderungen erfüllt** 2: **zwei Anforderungen erfüllt** 1: **eine Anforderung erfüllt**

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)

	- eine Beispielimplementierung im Projekt Viaduc ist vorhanden und einsehbar. **- Der Programmierer braucht sich nicht um GUI Aspekte zu kümmern.**	0: **keine Anforderung erfüllt**
Kriterium 5: **Entwicklerdokumentation ("Getting Started")**	Anforderungen: **- Bietet einen konzeptionellen Überblick** **- Schritt für Schritt Anleitung für Einbindung** **- Mind. 2 Beispiele für die Validierung** **- Mind. 2 Screenshots der resultierenden Darstellung im GUI**	Gütestufen: 3: **vier Anforderungen erfüllt** 2: **drei Anforderungen erfüllt** 1: **zwei Anforderungen erfüllt** 0: **eine oder keine Anforderung erfüllt**
Kriterium 6: **Systematik der Lösungsfindung/Lösungsvorschläge**	Anforderungen: **Ist es nachvollziehbar, warum gerade diese Lösung gewählt wurde?** **(was waren die Kriterien)?**	Gütestufen: 3: **Die Lösung ist sauber dargelegt. Es werden Alternativen verglichen** 2: **Die Lösung ist teilweise dargelegt. Alternativen sind erwähnt.** 1: **wenig Information, warum gerade diese Lösung bevorzugt wurde** 0: **Die Lösungsfindung ist nicht ersichtlich.**
Kriterium 7: **Codierung: Fehlerbehandlung**	Anforderungen: **Werden mögliche Fehler mit den entsprechenden Mitteln erkannt und behandelt?** **Mögliche Fehler: fehlerhafte Parameter, fehlende Parameter (vgl. Aufgabe)**	Gütestufen: 3: **eine Strategie zur Fehlerbehandlung wurde eingeführt und konsistent verwendet** 2: **Die Fehlerbehandlung ist lückenhaft / teilw. unbehandelt** 1: **Die Fehlerbehandlung wurde oft vergessen** 0: **Fehlerbehandlung nicht oder nur sehr rudimentär vorhanden.**

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)

Ausgangslage:

Vorgeschichte, Gründe, Umfeld:

CM Informatik entwickelt für die Schweizerische Bundesarchiv das Produkt "Viaduc". In diesem Projekt werden die Funktionen eines klassischen Lesesaals im Internet verfügbar gemacht.
Das Projekt ist gross und entsprechend komplex, denn es müssen viele existierende und neue, durch andere Anbieter zu erstellende Systeme zu einem funktionierenden Gesamtsystem integriert werden.
Es wurde eine MicroService Architektur gewählt, um die Komplexität im Griff zu behalten: Viele kleine Services mit klar definierten Aufgaben lösen zusammen eine komplexe Aufgabe.
Leider ist die Parameterverwaltung bei so vielen Microserices eine schwierige Sache. Da die Microservices möglichst autonom funktionieren müssen, haben wir zuerst jeden Service mit einer eigenen Konfigurationsdatei parametriert. Leider ist dieser Ansatz nicht benutzerfreundlich, weil erstens die Applikationsowner gar keinen Zugang zum Dateisystem der Server haben und zweitens die Konfigurationsdateien schon durch minimale Fehler im JSON-Format unbrauchbar werden.

Detaillierte Aufgabenstellung:

Die vom Kandidaten zu erstellende Programmmodule müssen den folgenden Personen dienlich sein:

- dem Programmierer, der bestimmte Parameter ins Programm einbauen möchte. Das sollte möglichst einfach vonstatten gehen. Es wäre zum Beispiel inakzeptabel, wenn der Programmierer für jeden Parameter ein GUI zeichnen müsste. Idealerweise muss er den Parameter nur benennen und beschreiben (Datentyp, Art der Validierung, Hilfetext, …) und den Rest erledigt das Programm. Der Einstieg in diese Art der Parameterverwaltung soll dem Programmierer möglichst leicht gemacht werden - jeder hat gerne glückliche Arbeitskollegen! Da diese aber manchmal unter Zeitdruck stehen, ist es von Vorteil, wenn man eine kurze Entwicklerdokumentation ("Getting started") schriftlich abgeben kann. Ebenfalls ist die Qualität des Codes entscheidend. Die Parameterverwaltung kann nicht immer verhindern, dass es fehlerhafte oder ganz fehlende Parameter gibt, aber sie muss immer auf diesen Missstand aufmerksam machen. Die Behandlung dieser Fehler wird im Rahmen der IPA abgehandelt. Nicht Bestandteil der IPA ist das Fehlerhandling innerhalb der anzusteuernden (teilweise bereits bestehenden) Services.

- dem Administrator, der die Parameterwerte der unterschiedlichen Microservices in einem einzigen GUI verwalten möchte. Hier spielt vor allem die Benutzerführung eine wichtige Rolle. Weil es sehr viele Parameter geben wird, muss man diese suchen können, mindestens anhand des Namens. Das System muss dann die vom Programmierer vorgegebene Validierung ausführen und gegebenenfalls eine sinnvolle Fehlermeldung anzeigen (z. B. formelle Richtigkeit des Parameters prüfen. Bsp: der Parameter e-mailadresse "hello world" wird als falsch erkannt, hingegen wird "muster.max@cmiag.ch" als valide eingestuft). Da es für viele Parameter sinnvolle Defaultwerte gibt, die man nur in Ausnahmefällen zu ändern braucht, sollen diese Defaultwerte angezeigt werden. Man sollte dann erkennen können dass dieser Wert der Defaultwert ist und man nicht

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)

selber einen anderen Wert eingetragen hat. Einige Werte müssen zwingend vorhanden sein, z.B. ein Datenbank-Connection String. Dann gibt es keinen sinnvollen Defaultwert. Das System muss zwingende, aber fehlende Parameterwerte dem Administrator signalisieren. Schliesslich muss der Parameter auch gespeichert werden können.

Die Speicherung der Parameter hat nicht zentral, sondern dezentral im jeweiligen Microservice zu erfolgen, in welchem den Parameter benötigt wird, damit die Microservices autonom bleiben und nicht auf einen funktionierenden Parameter-Dienst angewiesen sind.

Es ist wichtig, eine Lösung zu finden, welche beiden Benutzergruppen (Administrator und Programmierer) gleichermassen gerecht wird. Wir erwarten, dass verschiedene Lösungsvarianten im vorab durchstudiert werden. Dies gilt insbesondere für den GUI-Bereich: Hier ist es explizit erlaubt, sich von bestehenden Parameterverwaltungen von kommerzieller und nicht-kommerzieller Software inspirieren zu lassen - man braucht das Rad nicht neu zu erfinden. Auch in der Art, Weise und dem Zeitpunkt, wie und wann die Parameter an die Microservices übermittelt werden, gibt es verschiedene Lösungsvarianten mit Vor- und Nachteilen. Diese gilt es dann gegeneinander abzuwiegen und klug und nachvollziehbar zu entscheiden.

Beruht auf: (1) Verordnung des SBFI über die berufliche Grundbildung Informatikerin/Informatiker mit eidgenössischem Fähigkeitszeugnis (EFZ) vom 1. November 2013, (2) Ausführungsbestimmungen zum Qualifikationsverfahren mit Abschlussprüfung Informatikerin EFZ/Informatiker EFZ vom 1. November 2013 (Inkrafttreten am 1. April 2014)