

Aufgabe zur Gruppenarbeit im Modul: „Verteilte Systeme“ im Wintersemester 2021/2022

Ziel

Sie sollen ein Mashup entwickeln, welches verschiedene Kennzahlen rund um eine Kryptowährung übersichtlich zur Verfügung stellt. Hierzu sollen Sie unterschiedliche Quellen berücksichtigen und aggregieren.

Aufgabenbeschreibung

Ihre Aufgabe ist es, ein MashUp zu entwickeln und zu implementieren, das auf bereits bestehenden Diensten basiert.

Wenn Sie Ihr MashUp entwickeln, sollten die folgenden Anforderungen erfüllt sein:

- Es sollte verschiedene Datenquellen kombiniert werden.
- Ihre Lösung soll irgendwo gehostet werden, damit sie im Web zugreifbar ist.
- Sie müssen ein Backend-System implementieren, das zumindest Daten von zwei verschiedenen Diensten aggregiert und den architektonischen Ansatz verwendet, um mit der Same-Origin-Directive umzugehen. Diese aggregierten Daten sollten ebenfalls in dem MashUp visualisiert werden.

Das Mashup, das Sie implementieren sollen, wird ein Finanz-Dashboard sein. Hier sollen Sie Daten über eine Kryptowährung namens Waves integrieren. Der erste Dienst, den Sie integrieren sollen, sind die Marktpreise, die z. B. hier verfügbar sind:

<https://wavescap.com/info/api>

Hier implementieren Sie mindestens einen MA ([Moving Average](#)) und das [Sharpee Ratio](#) für das Paar Waves / USD (weitere Kennzahlen können Sie gerne implementieren). Zusätzlich sollten Sie die aktuelle Stimmung gegenüber der Kryptowährung Waves berechnen, indem Sie Tweets die folgende Hashtags / Benutzer beinhalten analysieren:

@waves.tech
@SignatureChain
@wavesprotocol
@neutrino_proto
\$Waves
\$USDN
@sasha35625

Sie können hierzu gerne fertige API's (sowohl im Web als auch als API für eine gängige Programmiersprache) zur Sentimentanalyse verwenden. Die Implementierung einer eigenen Sentimentanalyse ist nicht notwendig.

Zusätzlich sollten Sie Ihre eigene Idee umsetzen, wie Sie die auf [PyWaves](#) verfügbaren Zinsdaten für verschiedene Finanzprodukte auf der Waves-Plattform in Ihr Mashup integrieren können.

Sie müssen zunächst einige Benutzerszenarien definieren, die Ihre Ideen für die Implementierungen und die Anforderungen im Zusammenhang mit den von Ihnen vorgeschlagenen Konzepten veranschaulichen, bevor Sie diese entwerfen und umsetzen. Bitte nehmen Sie sich Zeit und überlegen Sie sich, wie Sie das MashUp entwickeln wollen. Im Rahmen der Dokumentation sollen Sie ebenfalls die Innovation Ihrer Lösung verargumentieren. Für die Umsetzung können Sie beliebige Web-Technologien, sowie gängige Programmiersprachen, verwenden (JavaScript, Python, Java, Angular 2, ReactJS, usw.)

Anforderungen

Ihre Einreichung muss mindestens die folgenden Dinge umfassen:

1. Sourcecode
2. Deployment URL der Services
3. Schnittstellenbeschreibung
4. Tests
5. URL unter der die Lösung deployed wurde
6. Beschreibung der Implementierung, inkl. Architektur

Ad 1: Den Sourcecode können Sie entweder in die ZIP Datei zur Einreichung packen oder über ein GIT Repository zugänglich machen.

Ad 2: Die entwickelten Services sollen deployed werden. Die hieraus resultierenden URL's geben Sie bitte in Ihrer Dokumentation an.

Ad 3: Eine Beschreibung der Web Service Schnittstellen, gerne mittels Swagger (<https://swagger.io/>), muss ebenfalls beinhaltet sein.

Ad 4: Die in Ihren Web Service implementierten Methoden müssen mittels Unit Tests getestet werden.

Ad 5: Im Rahmen der Bewertung wird die Implementierung getestet, hierzu geben Sie bitte die URL der deployten Lösung in der Dokumentation an.

Ad 6: Sie erstellen eine Dokumentation der Lösung, des Vorgehens und der Architektur. Hierzu verwenden Sie bitte das IEEE Format. Die Länge der Dokumentation darf 5 Seiten im DIN-A4-Format nicht überschreiben, dies beinhaltet alle Angaben, Literatur, Anhänge, ...

Entsprechende Formatvorlagen finden Sie hier:

<https://www.ieee.org/conferences/publishing/templates.html>

Bewertungsschema

Das Bewertungsschema wird folgende Punkte berücksichtigen:

Kriterium	Anteil
Qualität des eingereichten Codes	25%
Qualität der Schnittstellendokumentation	5%
Testabdeckung	10%
Dokumentation	20%
Korrektheit und Vollständigkeit der Lösung	40%