Running Web Assembly Smart Contracts in Fabric

Title	Running Web Assembly Smart Contracts in Fabric
Status	**PROJECT COMPLETED**
Difficulty	**MEDIUM**

Description

WebAssembly (WASM) has slowly become a popular language for smart contract development. One of the key features of the WebAssembly is that different languages such as Rust, Python, C/C++ support compilation to WASM. By integrating WASM into Hyperledger Fabric, chaincode developers will be given a wider variety of languages to choose from when constructing their smart contracts.

The overall goal of this project is to enable this feature in Fabric. We are looking for an intern who is interested in adding this support. He/she will be tasked to come up with a design for the integration and achieve a basic implementation by the end of the internship. As well as integrating WASM, we are also looking for the development of examples as to how this new feature will be used in the workflow the intern creates. We have provided a possible Go implementation of WASM as a possible resource/library, however we are open to the intern choosing a different implementation and chaincode language.

Additional Information

- https://github.com/hyperledger/fabric - want functionality for Fabric
- https://github.com/perlin-network/life - suggested WASM vm to be integrated. Other WASM implementations can be used instead.
- https://github.com/hyperledger/fabric-chaincode-evm - example of the potential of WASM/how wasm can be integrated

Learning Objectives

- Learn to work in open source development which includes communicating using open source channels, working with the community to get feedback and help to complete the project
- Learn good software development practices
- Learn to communicate and work global teams
- Learning to design a project/feature end to end

Expected Outcome

- Evaluations of the multiple web assembly libraries for a language and why we should base out implementation on any specific library.
- A chaincode that once installed in Fabric can be used to run Web Assembly Smart Contracts
- Sample applications/tutorials to go along to use the feature. As a user of this feature I should be able to follow this tutorial to understand end to end.

Relation to Hyperledger

We want to add this feature to Hyperledger Fabric (https://github.com/hyperledger/fabric).

Education Level

Students in undergraduate programs can apply.

Skills

- Git understanding of source control
- Go (Preferred), if not then comfortable with languages such as C/C++, or Java
- Familiarity with unix commands

Future plans

Based on the popularity of WASM we would like to turn this into project/feature we continue and develop in a similar fashion as the fabric-chaincode-evm.

Preferred Hours and Length of Internship

Both full time and part time are okay.

Mentor(s) Names and Contact Info

Usernames below are for chat.hyperledger.org

Morgan Bauer, mbauer@us.ibm.com, @MHBauer, IBM
Jay Guo, guojiannan@cn.ibm.com, @guoger, IBM
Swetha Repakula, srepaku@us.ibm.com, @swetha, IBM

Mentor

shubham aggarwal

Frequently Asked Questions

Q: Why golang?

A1: We have already written one chaincode in golang (see above). Thus we would be most helpful to guide writing another chaincode in golang.

A2: It must be in a supported chaincode language, right now that is Java, Javascript, and golang.

Q: What should I do to prepare?

A: There should be time to preparation and learning during the actual internship. If you want to get a taste for the coding part of what you'll most likely be doing, check out the fabric-chaincode-evm project linked above. Try and follow the tutorials and deploy it. Make a change and redeploy it.

Project Deliverables

Week 1-2	☑ Research and come up with a high-level strategy to integrate WASM with hyperledger fabric.
Week 3-4	☑ Evaluate and decide on which WASM VM to integrate
Week 5-6	☑ Chaincode creation (wasmcc) for integrating WASM VM.
Week 7-8	☑ Create and implement a strategy to store the external wasm chaincodes in the state.
Week 9-10	☑ Expose SHIM layer function in wasmcc chaincode, which can be imported by wasm chaincode developers.
Week 11-12	☑ Transaction to instantiate a wasm chaincode, i.e., retrieve from state and instantiate it in wasm vm.

Week 13-14	☑	Export functions from wasmcc for wasm chaincode developers
Week 15-16	☑	Define function definition which needs to be implemented by wasm chaincode developers, which will be executed by wasmcc at the time of deployment for every new webassembly chaincode modules.
Week 17-18	☑	Define "init", "query" & "invoke" function definition which needs to be implemented by wasm chaincode developers.
Week 19-20	☑	Write sample chaincode and compile to wasm, and deploy it in a sample hyperledger fabric network through wasmcc.
Week 21-24	☑	Implement an end to end use case using wasm chaincode.

Project Repository:

https://github.com/hyperledger-labs/fabric-chaincode-wasm

Project Summary :

Project Fabric c...aincode wasm.pdf

Quick Demo Video:

Your browser does not support the HTML5 video element