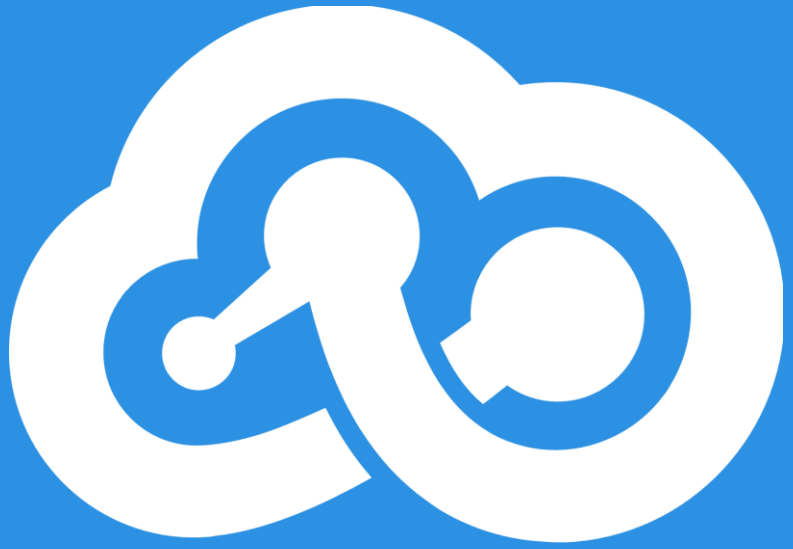


Blockchain | Cryptlets -Next Generation Oracles

Cale Teeter
DX/TED - SDE

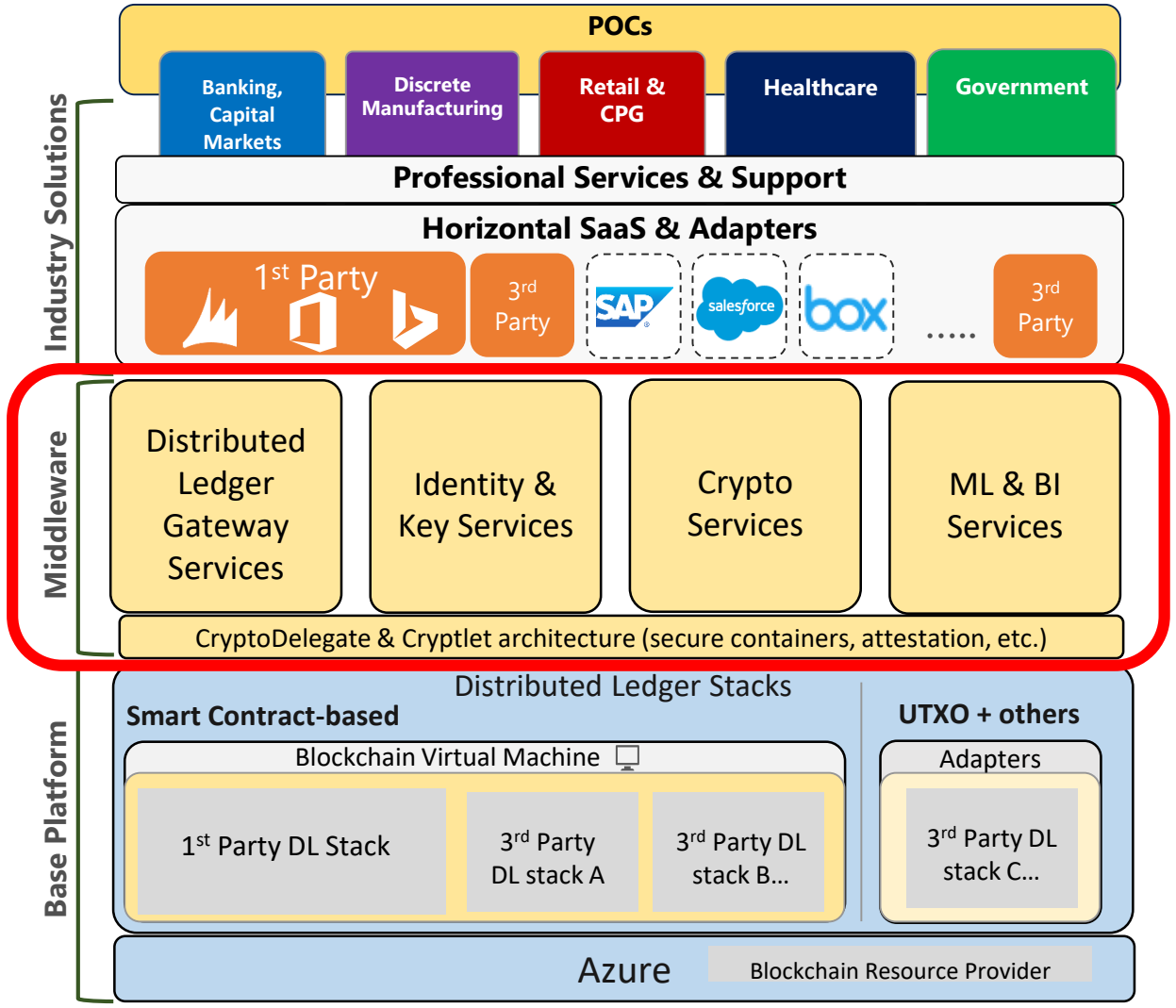
December 2016

Microsoft will execute on its strategy in three steps: Learning from POCs, growing the ecosystem, and building key middleware

1 - Build and learn from key customer-driven **POCs** built on top various blockchain technologies

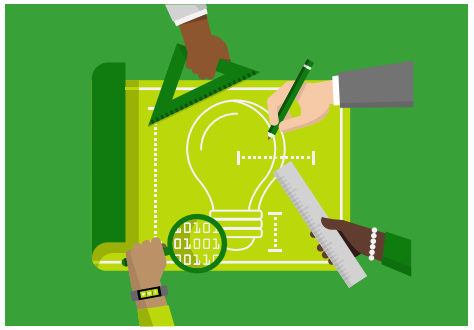
2 - Grow the blockchain **marketplace ecosystem** & artifacts together with our partners & customers

3 - Develop key Azure blockchain **middleware services** to ensure the infrastructure is enterprise ready

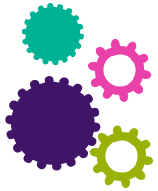


Bletchley: Open Infrastructure, Enterprise Capabilities

Data Services



Solutions



Existing Systems

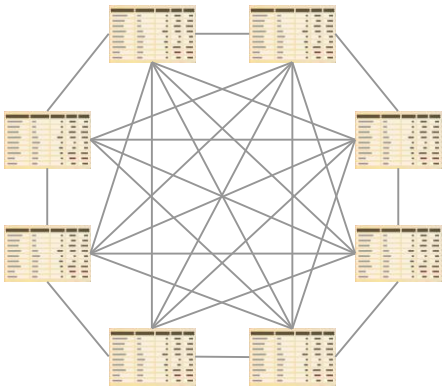
Identity

Key Management

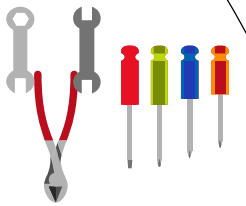
Security In Depth

Privacy

Operations & Management



Blockchain has some missing parts…



Better Tools

Current Limitations

Limitation	Example
No notion of real world time	Do something @ specific Time i.e. 4:00 PM EST Do this every 5 minutes
Can’t react to real world events directly	Do this when oil hits $40 a barrel
Code execution scaling is not straight forward	I need this to run FAST
Hard to implement libraries, versioning is DLL Hell	How do I get code reuse? I need to version a referenced SmartContract
Code in the clear in all cases	My algorithm is company IP
Trusting and using external code or data is dangerous	How can I trust this data hasn't been tampered with? Is this code running in isolation?

Need to have Trusted External Data?

- Receive Market Data based on an event?
  - Specific Time i.e. 4:00 PM EST
  - Specific Interval i.e. every 15 minutes
  - Price of something hits a threshold i.e. Oil goes above $40 a barrel
- Receive data based on external application updates, i.e. CRM System customer credit rating drops
- Request and Receive results from a High Performance Computing job i.e. Monte Carlo simulation, Gene Sequence complete, etc.

Need to have Trusted Execution?

Need secure execution in completely secure isolation and attested that it was not tampered with during execution?

- Secure IP protected algorithms but still share with the blockchain network: i.e. derivative pricing algorithm that multiple counter parties agree to use for a contract, but the actual algorithm remains secret, but attested.
- Scale an algorithm for maximum performance by running it off the blockchain in a secure and attested way.
- Perform complex interactions like distributed transaction coordination across many systems in a secure way.
- Use libraries for common platforms like Java and .NET in your SmartContracts.

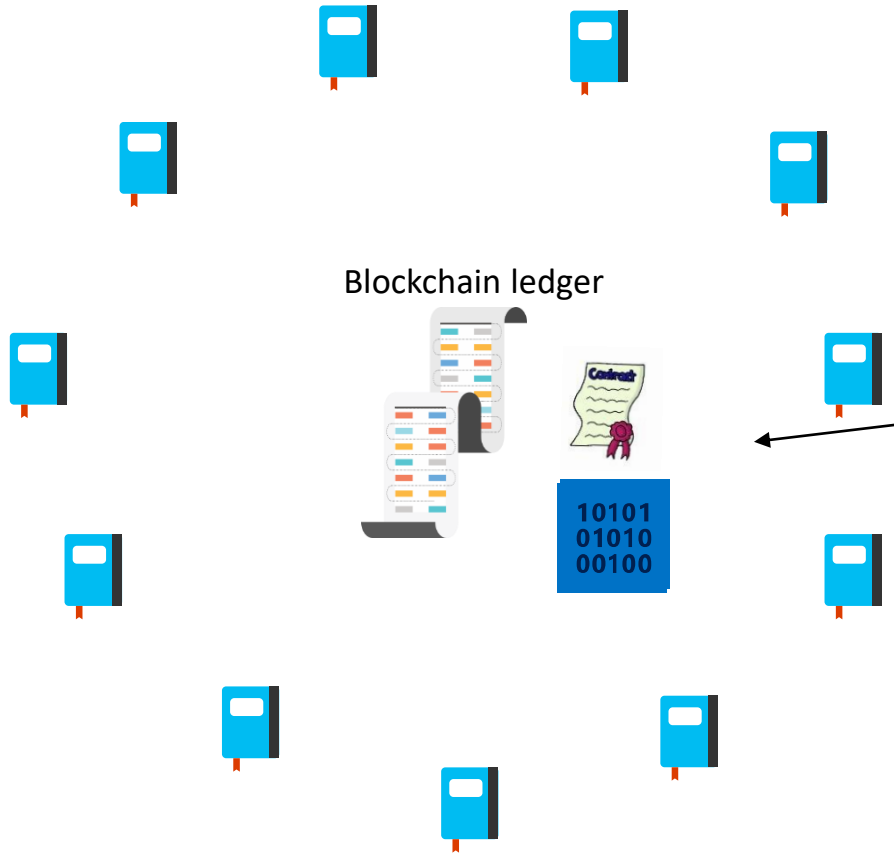
Introducing Cryptlets – Secure Distributed Middleware

Bank

A bank, hedge fund and insurance company enter into a SmartContract

Hedge Fund

Insurance

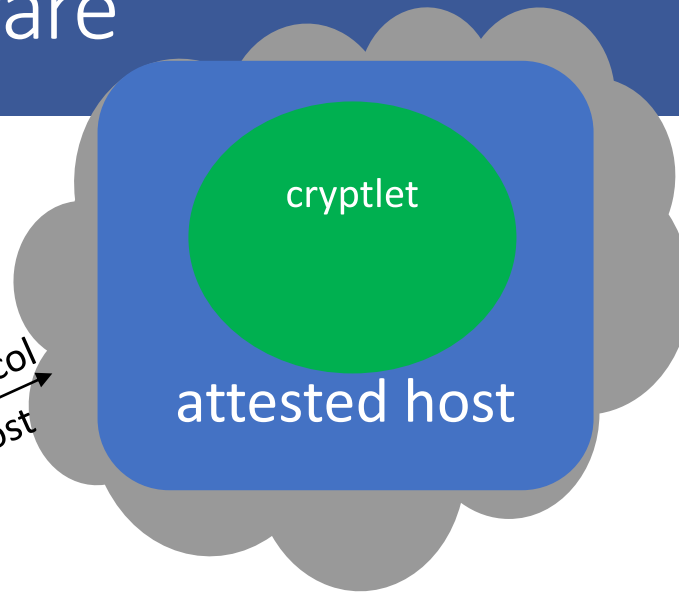


Blockchain ledger

Virtual machine

(Node)

Cryptlets



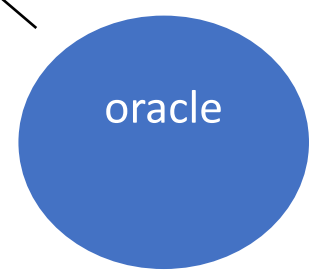
Secure protocol

Trusted host

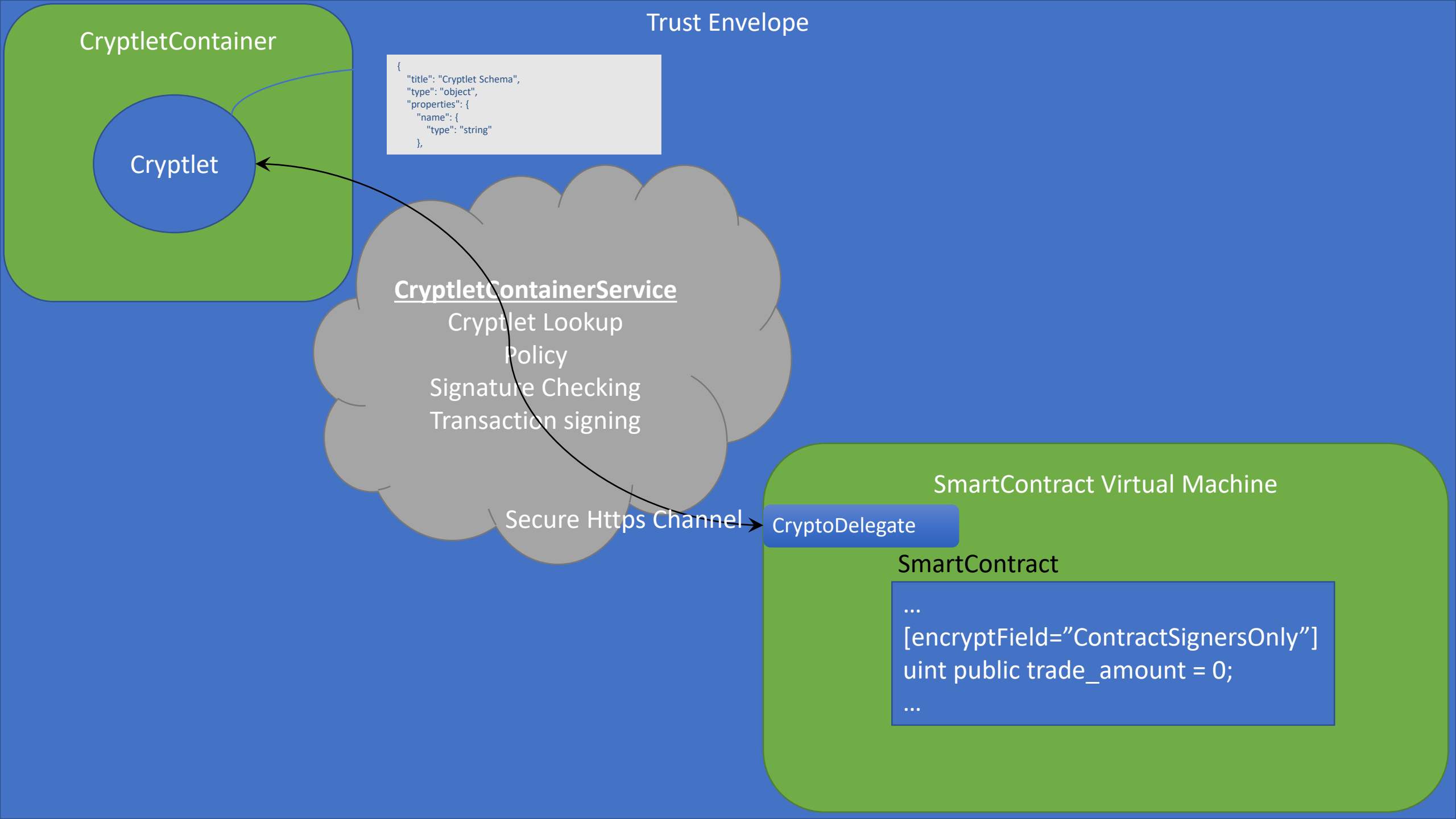
Today

Secure protocol?

Secure host?



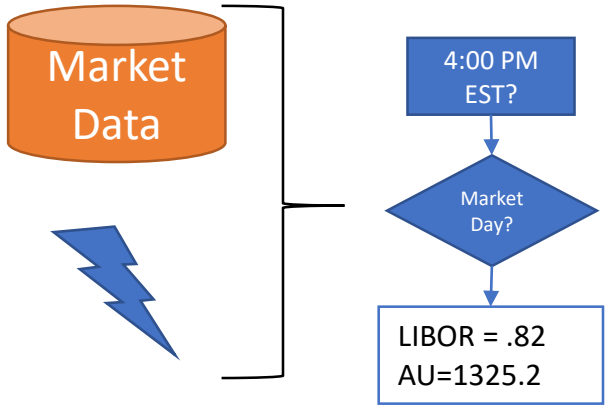
Everyday at 4 PM EST it needs a calculated rate like:
(LIBOR * .04%) + Diff(Gold)

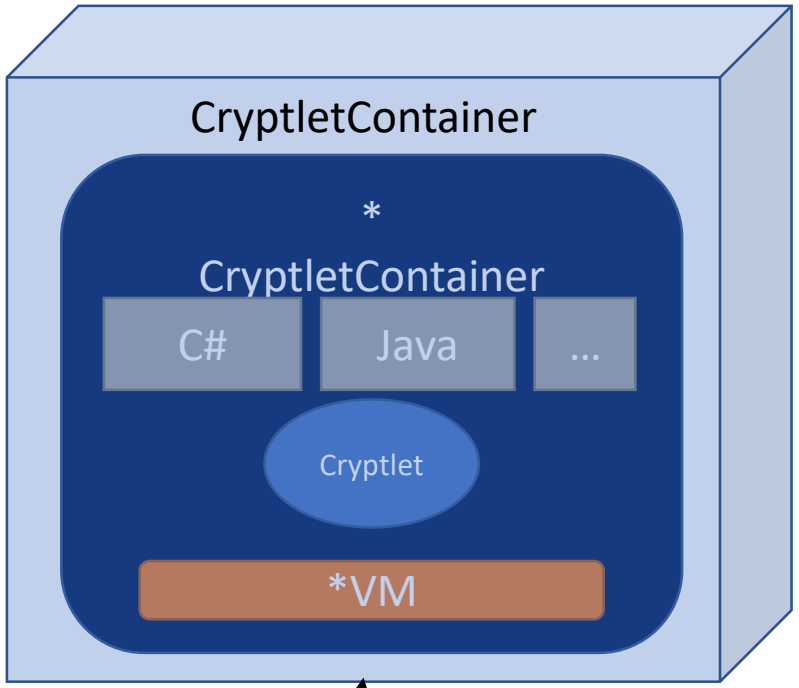

Trust Envelope
CryptletContainer
Cryptlet
{
"title": "Cryptlet Schema",
"type": "object",
"properties": {
"name": {
"type": "string"
},
CryptletContainerService
Cryptlet Lookup
Policy
Signature Checking
Transaction signing
Secure Https Channel
SmartContract Virtual Machine
CryptoDelegate
SmartContract
…
[encryptField=”ContractSignersOnly”]
uint public trade_amount = 0;
…

Cryptlet vs. oracle

Cryptlets	oracles
(+)Trust with Verification – trust hoster (HTTPS), trust Cryptlet key & trust enclave signature	(-)Requires trust but no formal verification
(+)Standard Infrastructure - Hardware based isolation and attestation via enclaves available Globally in Azure	(-)Custom – write & host separately and establishing trust difficult
(+)Integrated developer use with Aspects and tooling	(-)Custom – write your own
(+)Marketplace for publishing and discovery	(-)No common marketplace, no publishing or discover tools
(+)Bletchley Cryptlet SDK frameworks to get started quickly creating and consuming Cryptlets (Utility, Contract)	(-)Platform specific, documentation sparse
(+)Multiple language options as well as blockchain agnostic	(-)Custom

Use Case - Event





NODE
SmartContract Virtual Machine

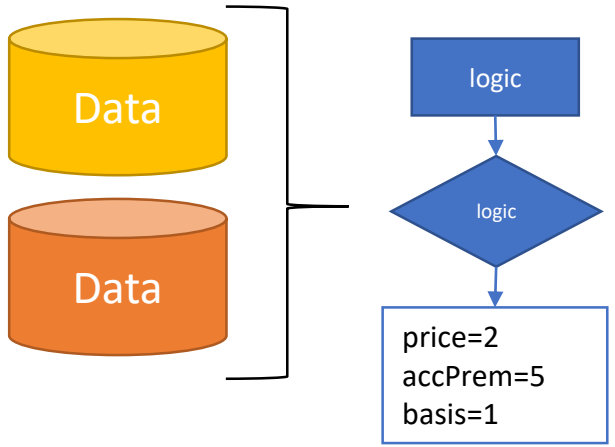
CryptoDelegate

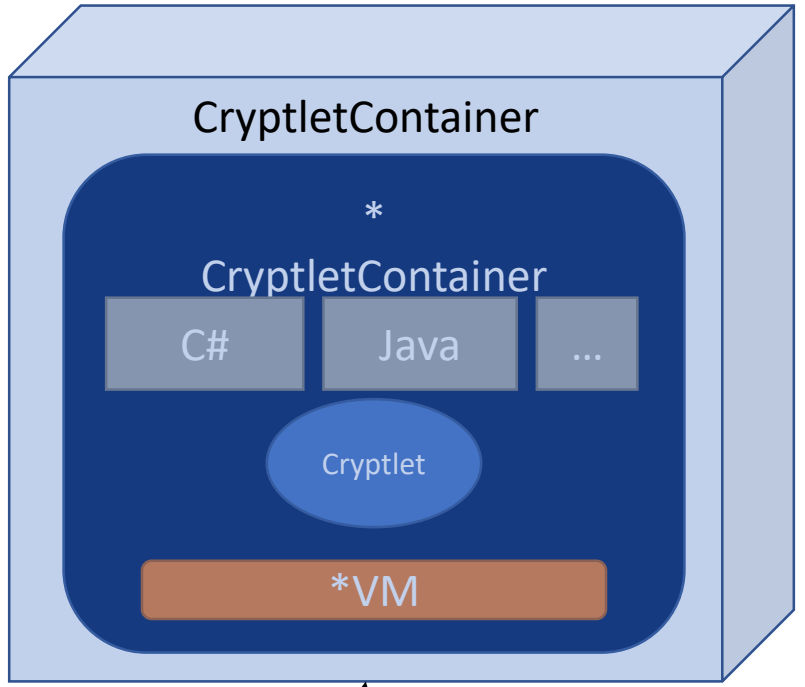
SmartContract

```
function init(){
  var marketWatcher = MarketWatcher();
}
//callback method that is run when the Cryptlet PriceUpdate event fires
[event(marketWatcher.MarketEvent('16:00 GMT-5', true, 'LIBOR', 'AU')];
function CalculatePrice(var eventData){
...
}
```

Event Subscription

Use Case - Control





NODE
SmartContract Virtual Machine

CryptoDelegate

SmartContract

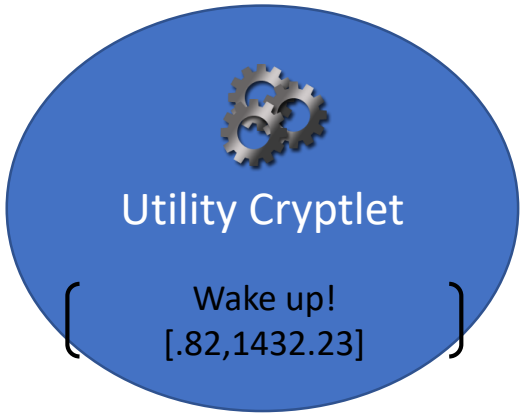
```
contract CreditDefaultSwap is ContractCryptlet{
   uint price;
   uint accPrem;
   utin basis;
//properties
    function Price(uint val){
       price=val;
   }
    function AccPrem(uint rate){
```

price	accPrem	basis	...
2	5	1	...

Control Surrogation

Microsoft BaaS | Utility Cryptlet

Cloud



Blockchain Node

SmartContract

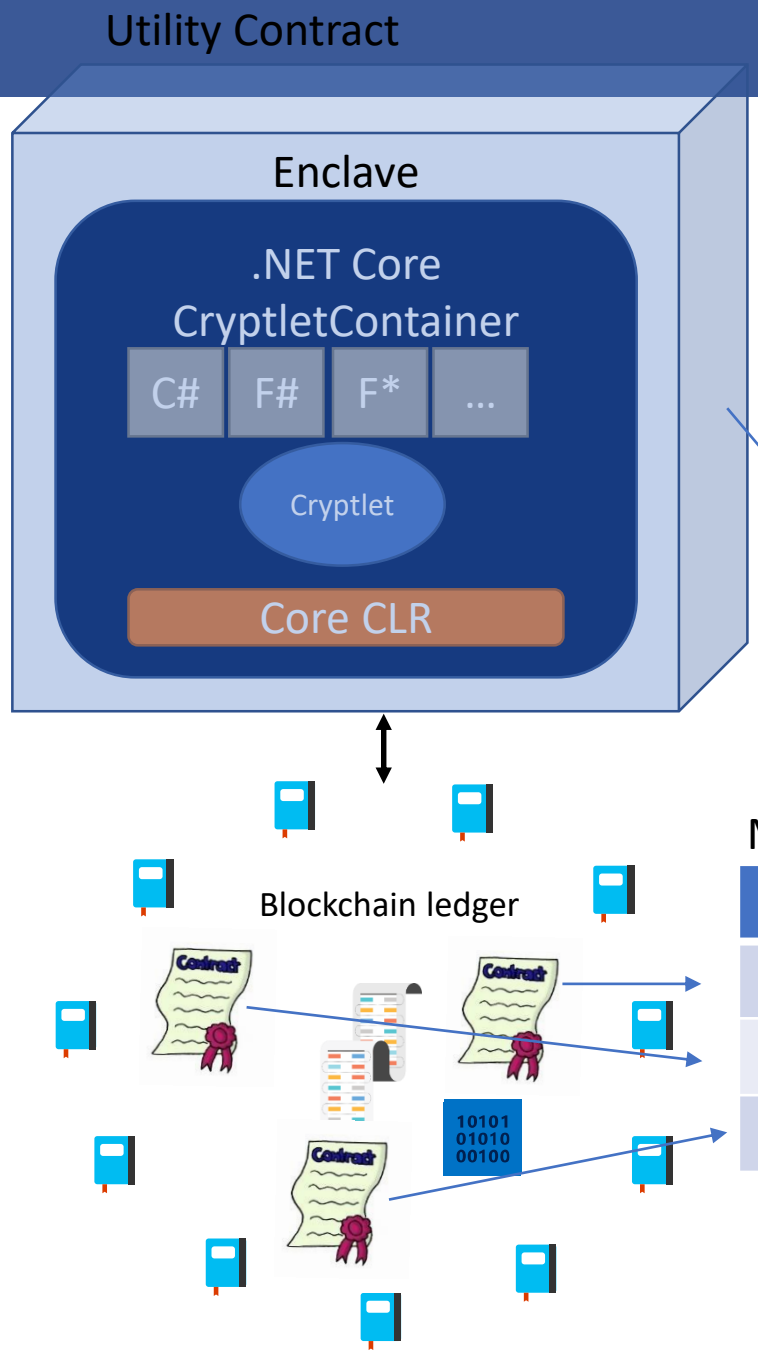
Subscribe: 4PM EST, Markets Open, give me LIBOR and Gold

```
function init(){ //or function smartContractName() as the constructor
    [event(stockclient.PriceUpdate.Subscribe('16:00 GMT -5", true, 'au', CalculatePrice)];
    stockClient = StockClient();
}

function CalculatePrice(var prices){
       user owner = userList[msg.sender];
       owner.exists = true;
       owner.balance = 100000000000;
       owner.role = ROLE_DEALER;
       CPIDCOUNT = 1;
       TOCOUNT = 1;
       standardTerms.ticker = "GE CP";
       standardTerms.quantity = 1;
       standardTerms.par = 10000000000; // $ / 10000  --> Written in tenthousandth's of a
       dollar (more precision because calculated amounts will be in this.)
       standardTerms.maturitylength = (30*24*60*60)/TD;
       standardTerms.discount = 735; // % / 100  --> Written in hundrendths of a percent
       (less precision allowed since this value does not get operated on)
       Trade_amount = ((standardsTerms.quantity/stardardTerms.quantity) * rate >
              {ROLE_DEALER].discount %* TOCOUNT++);
       …
}
```

CryptoDelegate

Message Queue Pattern

1. Message Queue SmartContract subscribes to the Utility Cryplet to receive price updates every 2 hours when market opens for MSFT, BAC & AU
2. Other SmartContracts can simply look up prices both most recent and historical from the Message Queue SmartContract



Message Queue SmartContract

ticker	price	hash
msft	56.23	0x2e423…
bac	12.23	0x4df21…
au	1522.12	09ce233…

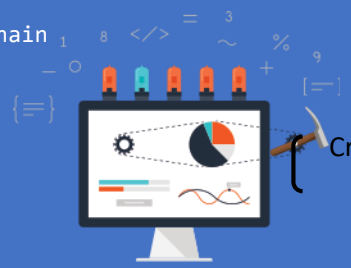
Microsoft BaaS | Contract Cryptlet

Blockchain Node

SmartContract

Deploy CreditDefaultSwap

```
//functions written in C# for Contract Cryptlet
import “github.com/cryptlets/swaps/cds.cs” as code;

contract CreditDefaultSwap is ContractCryptlet{

 //state stored by SmartContract in blockchain
 uint public trade_amount = 0;
 uint price;

 //SmartContract Constructor

 Function MySmartContract(){

  _code=code;

 }
}
```



Create Contract Cryptlet

Written to Blockchain

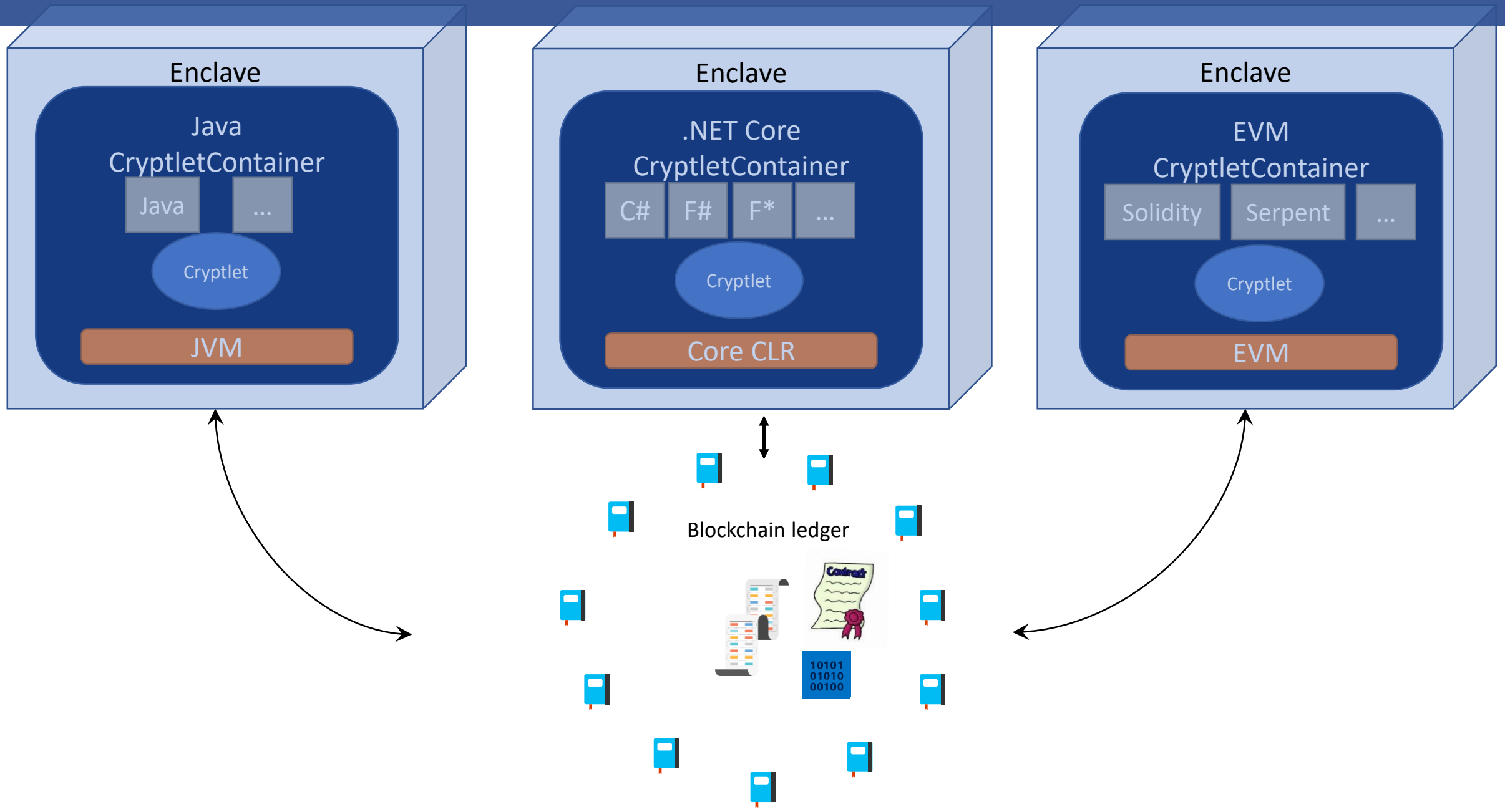
Cloud

Contract Cryptlet

trade _amount = 22.42;
price=*encryptedValue*;

CryptoDelegate

Secure Execution and Secure Data – Contract, Control and Encryption Services



Lots of Infrastructure

Requirement	Blockchain Fabric
How to use Cryptlets?	Aspects via code tags for behavior
Interpretation of aspects and validation of cryptlet communications	CryptoDelegate registers behaviors and inspects Cryptlet to SmartContract communications
Discovery and Management of Cryptlet Fabric	CryptletContainerService + Azure Service Fabric
Secure Data and Execution	CryptletContainer + Enclaves
Key Management and Lifecycle	Identity and Key Management Service + Azure KeyVault
Advanced encryption services – ECC, zkP, ring, threshold, etc.	Key Management Service and Encryption Cryptlets
Discover, Register and Use Cryptlets	Azure Cryptlet Fabric

Links

- Project Bletchley – Whitepaper - https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/bletchley-whitepaper.md
- Project Bletchley – The Cryptlet Fabric - https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/CryptletsDeepDive.md
- Azure Blockchain Marketplace - https://azure.microsoft.com/en-us/marketplace/?term=blockchain
- Intel SGX - https://software.intel.com/en-us/sgx