A Lap around Cryptlets

Bletchley

Marley Gray
Principle Architect – Program Manager
Azure Blockchain Engineering

September 2016

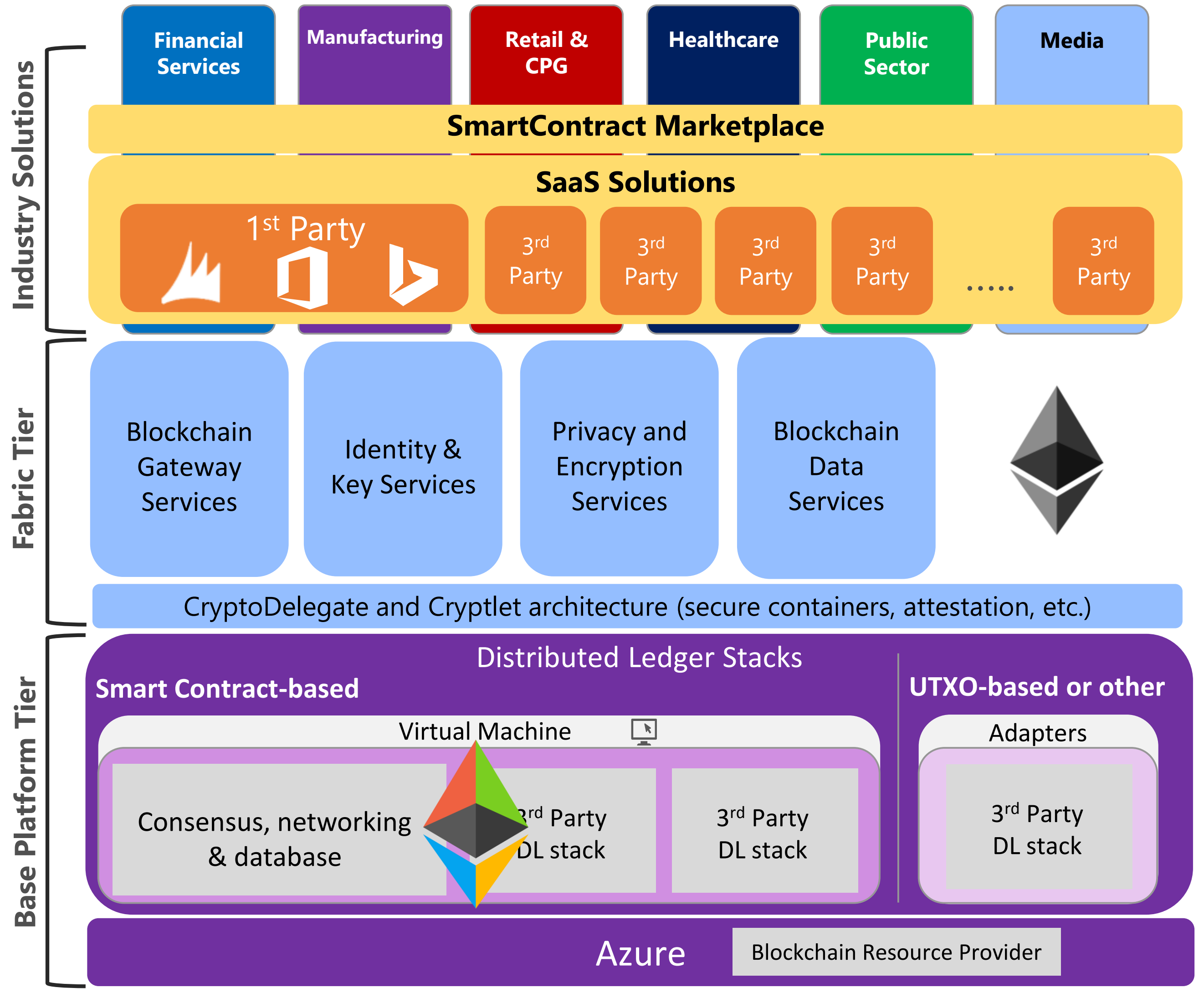
Microsoft

Microsoft & Ethereum

- First for both platforms – started together
- DevCon1 & 2 sponsor
- Ethereum is and will continue to be a 1st class citizen on Azure
- Support for community and partners – BizSpark, Meetups and Workgroups (Kinakuta)

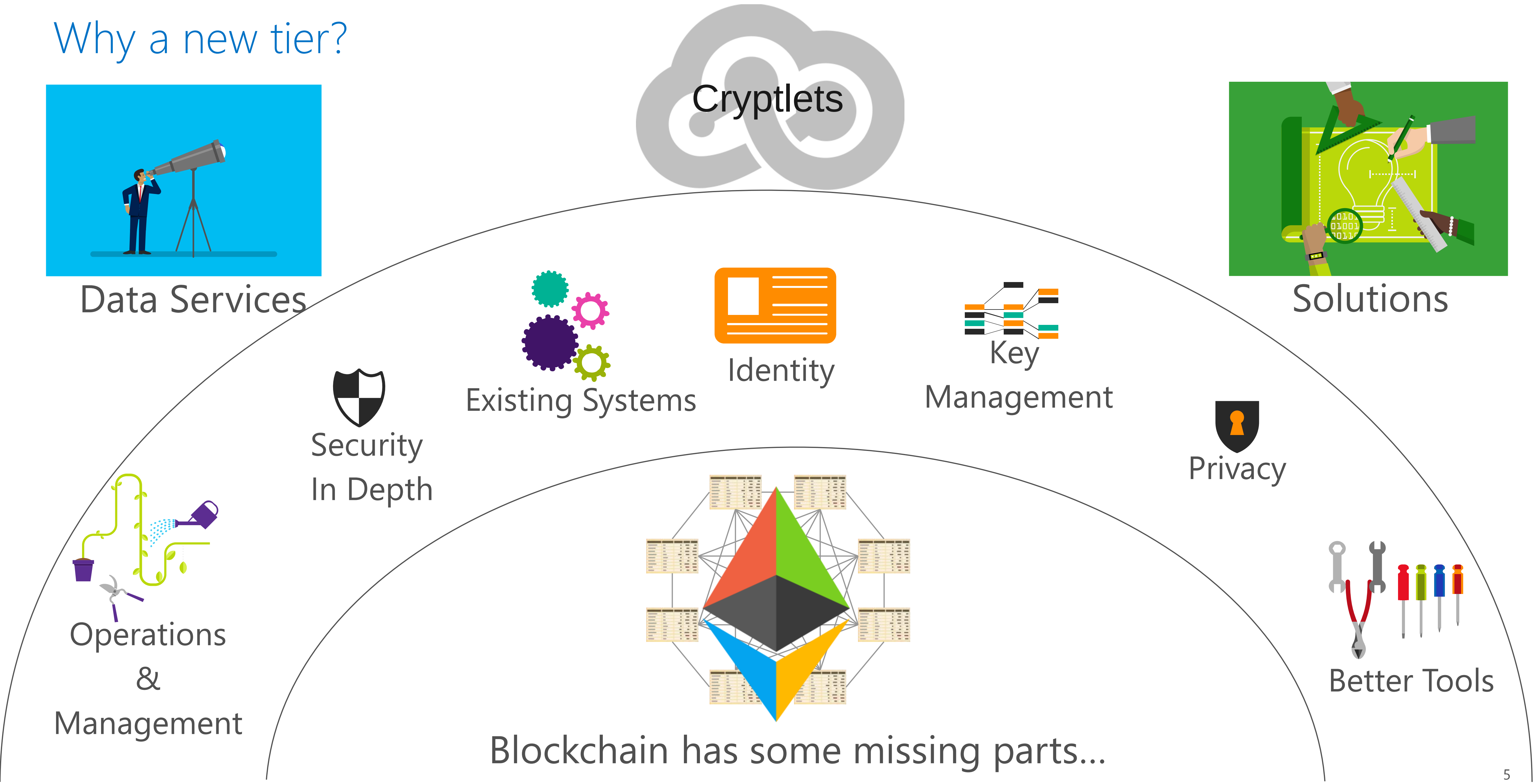
Announcing…

Bletchley v1



Lap around Cryptlets

Why a new tier?



More than just Trusted Data…but Execution?

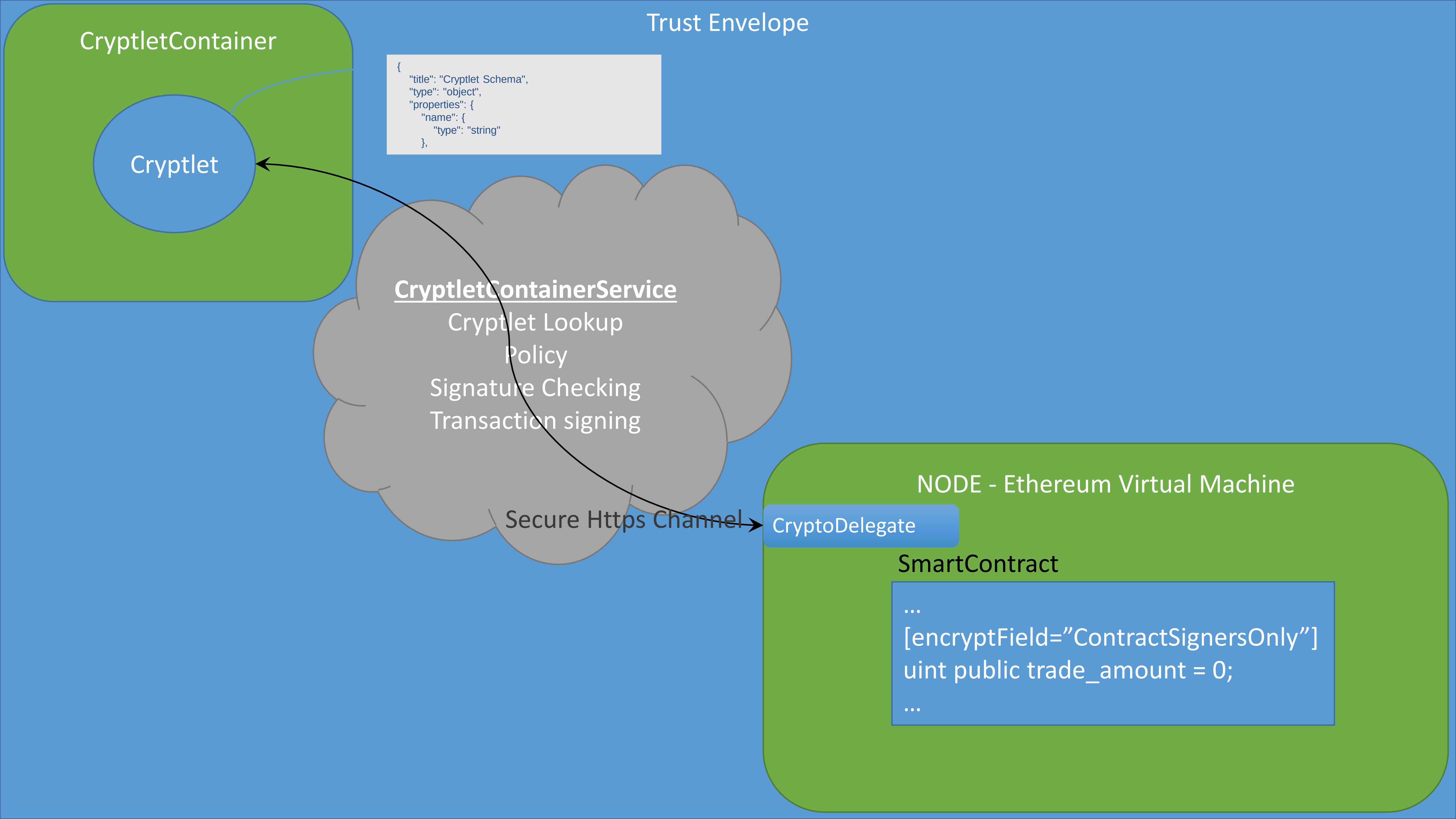
- Receive Market Data based on an event?
  - Specific Time i.e. 4:00 PM EST
  - Specific Interval i.e. every 15 minutes
  - Price of something hits a threshold i.e. Oil goes above $40 a barrel
- Secure IP protected algorithms but still share with the blockchain network: i.e. derivative pricing algorithm 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

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 (SGX) 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

Lots of Infrastructure

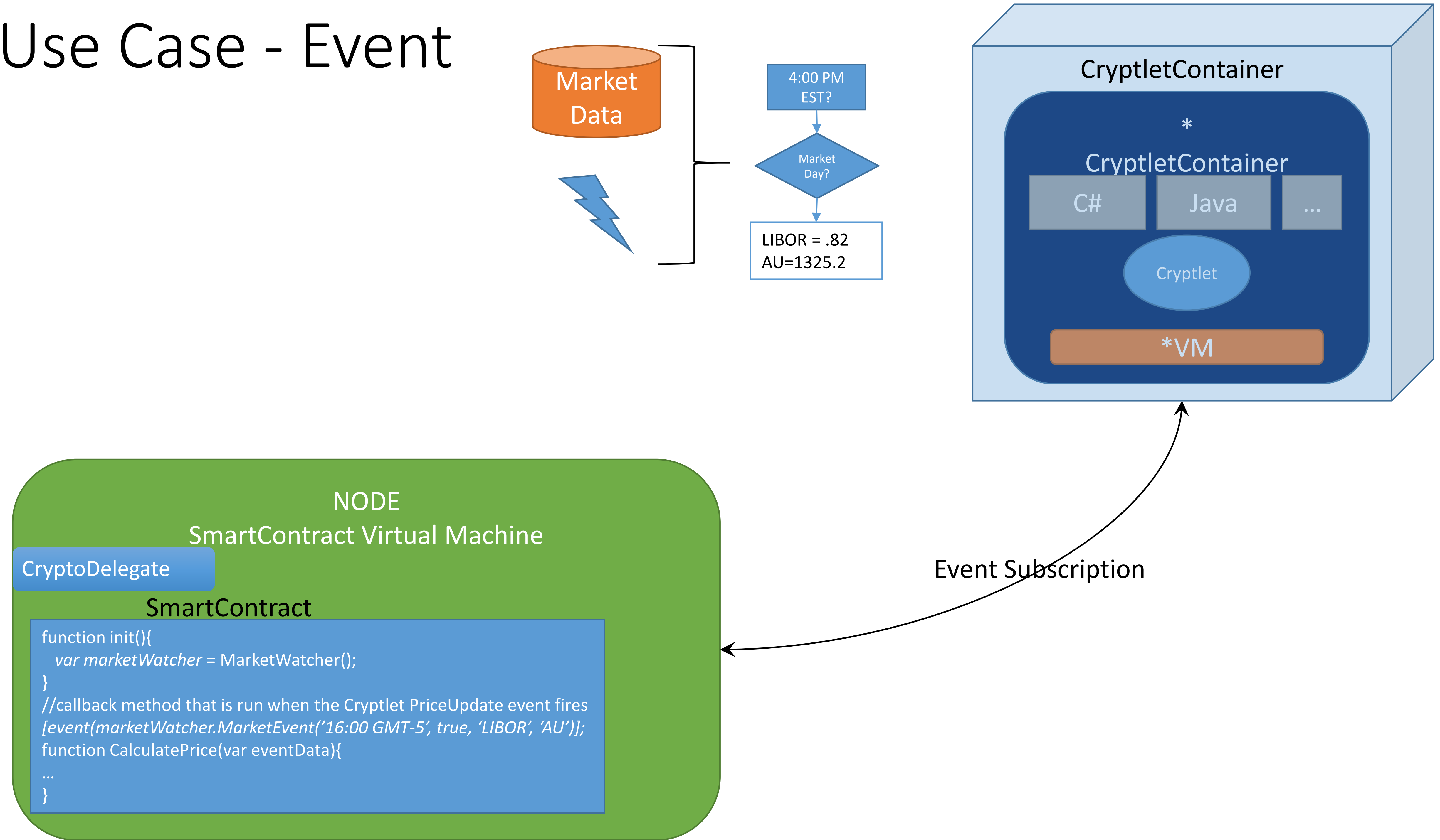
Requirement	Blockchain Fabric
How to use Cryptlets?	Aspects via code tags for behavior
Interpretation of aspects and validation of cryptlet communications	CryptoDelegate registers behaviors and inspects Cryptlet to SmartContract communications
Discovery and Management of Cryptlet Fabric	CryptletContainerService + 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 + Marketplace



```
{
    "title": "Cryptlet Schema",
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
```

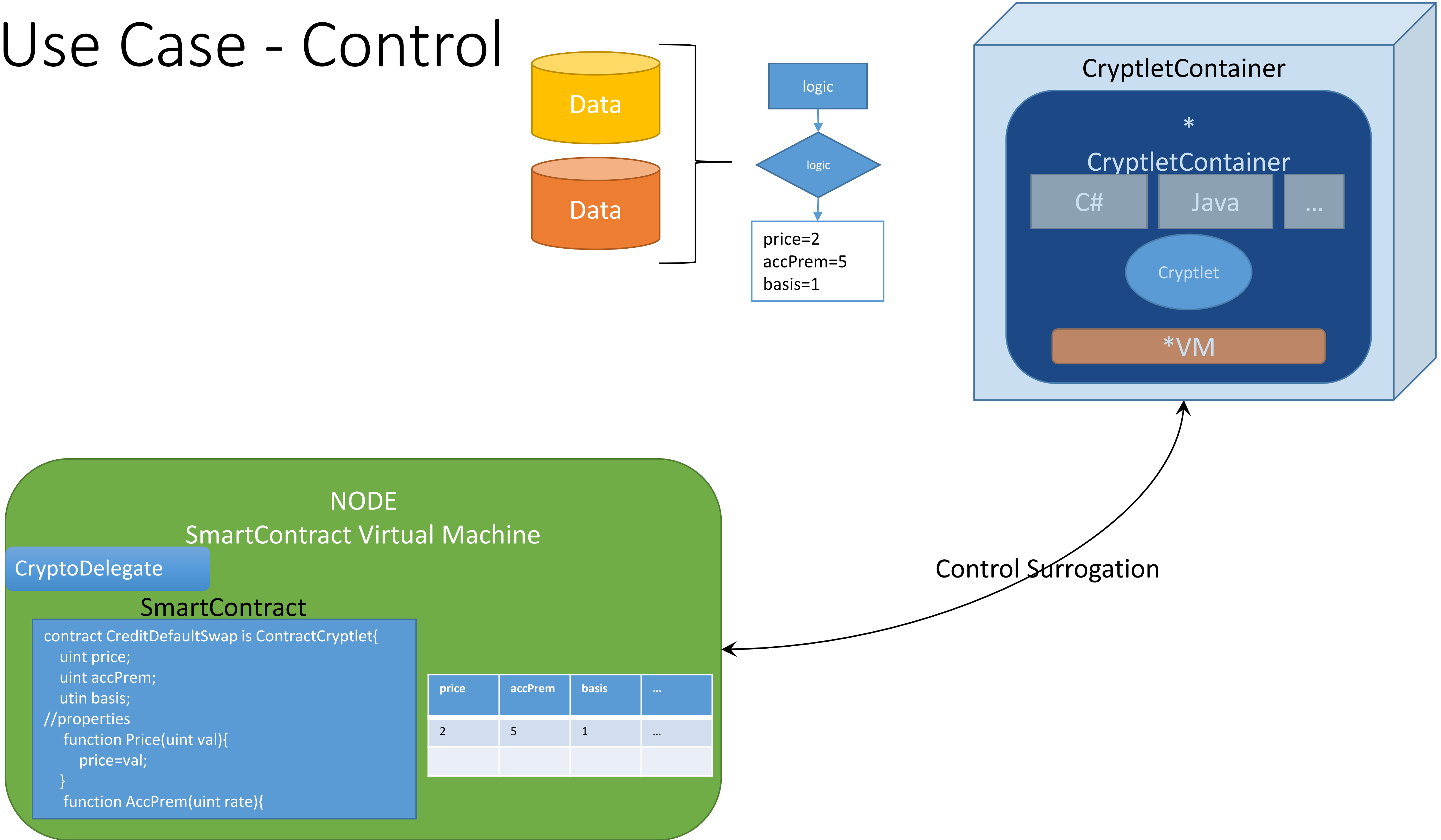
```
…
[encryptField="ContractSignersOnly"]
uint public trade_amount = 0;
…
```

Use Case - Event



```
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){
…
}
```

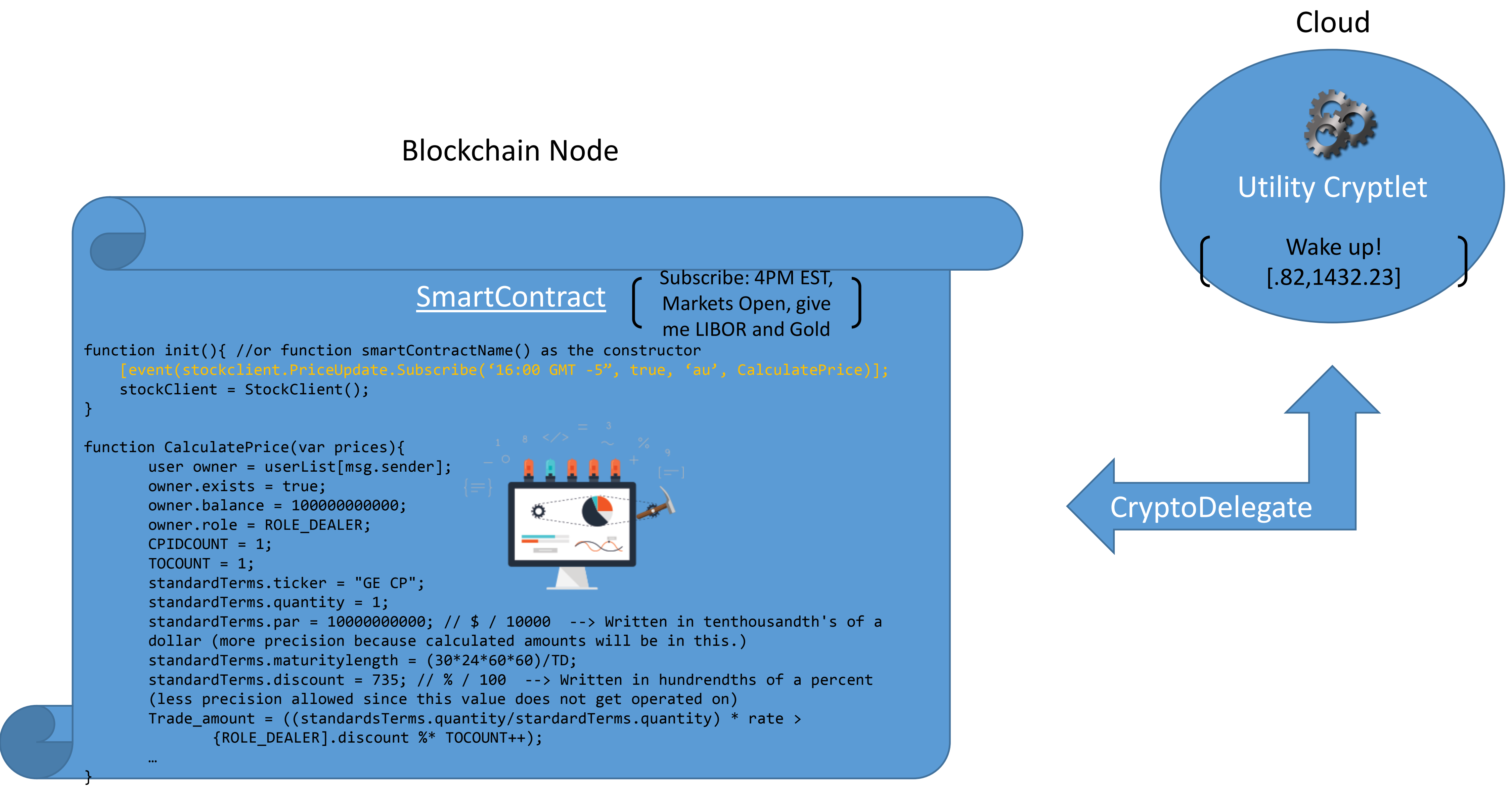
Use Case - Control



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

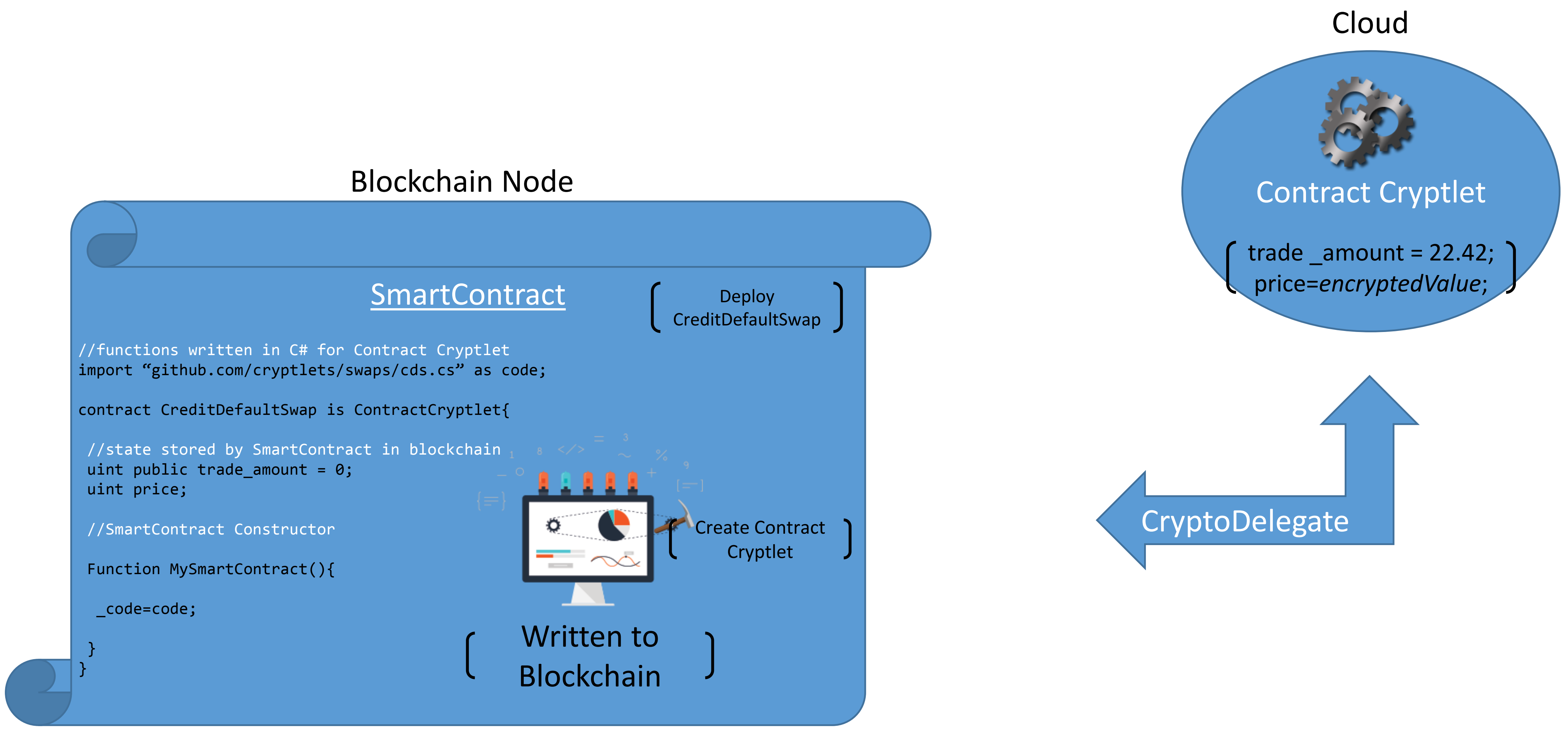
price	accPrem	basis	...
2	5	1	...

Microsoft BaaS | Utility Cryptlet



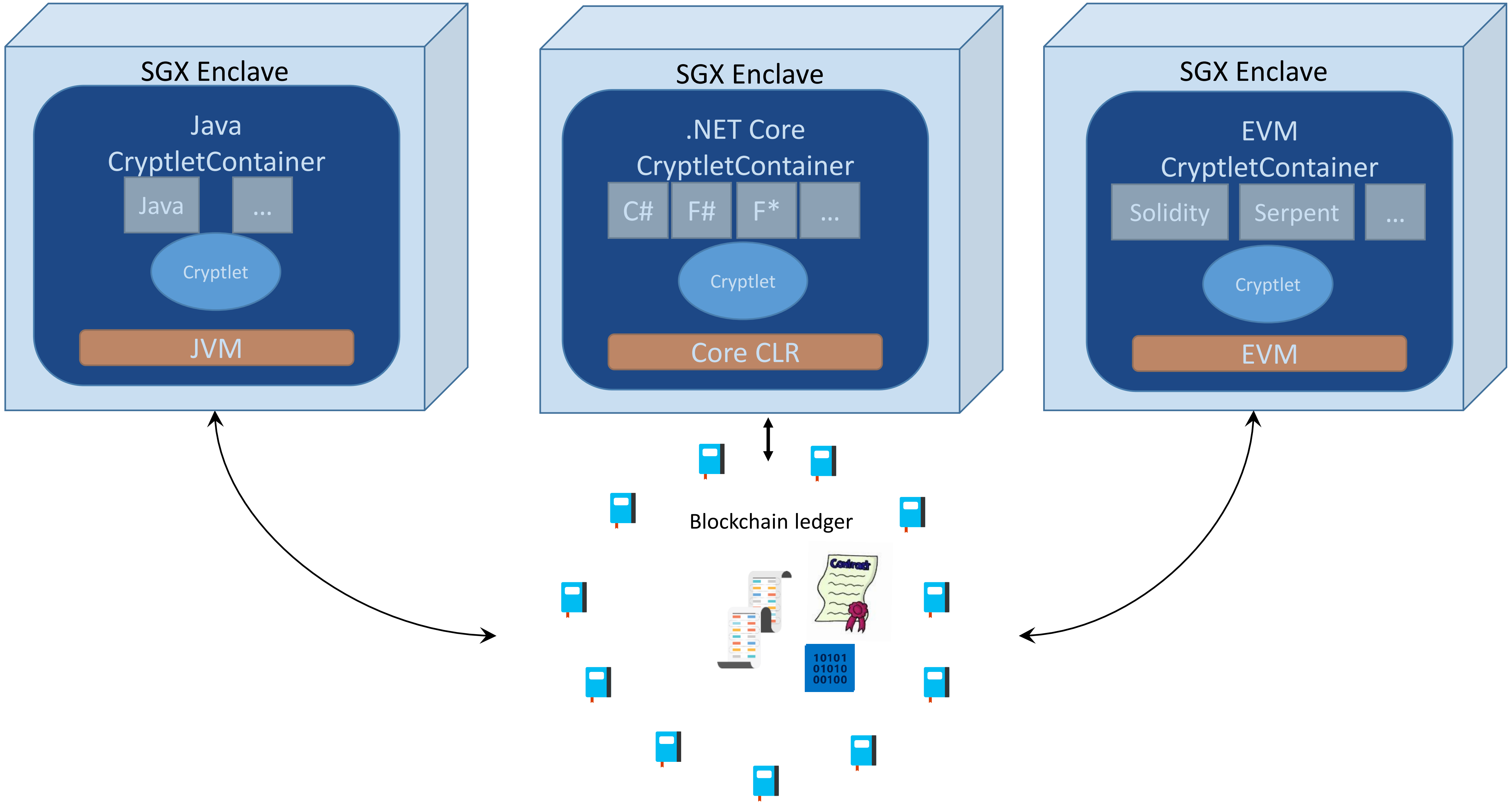
```
function init(){ //or function smartContractName() as the constructor
    [event(stockclient.PriceUpdate.Subscribe(‘16:00 GMT -5”, true, ‘au’, CalculatePrice)];
    stockClient = StockClient();
}

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++);
       …
}
```

Microsoft BaaS | Contract Cryptlet



```
//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;

 }
}
```

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



5 Points - World Wide Hyper Scale Blockchain Application Fabric

- Secure Execution with Enclaves on demand
- Secure Data Providers with end to end attestation
- Scalability and Flexibility in code execution
- Developer Friendly discoverability and use broad ecosystem
- Standard way of publishing and accessing external resources

Bletchley v1

Ethereum Consortium Blockchain Network

https://azure.microsoft.com/en-us/documentation/templates/ethereum-consortium-blockchain-network/

Pre- ARM Template

3 weeks

To set up a mock consortium network in Azure today:

1. Review public Ethereum network **documentation**
2. Determine **topology** for a consortium network
3. Map topology to **Azure resources** (VMs, Storage Accounts, etc.)
4. Write **ARM template** or script OR manually deploy
5. Configure Ethereum client via **Linux BASH scripts** to support private network (peering, isolate mining nodes, etc.)
6. Configure other **Ethereum protocol properties** (genesis block, max peers, etc.)
7. Set up **Ethereum accounts** and allocate **ether**
8. **Trial and error** to make above steps work
9. Integrate with other **Azure services**, such as AAD and Key Vault
10. **Test** template

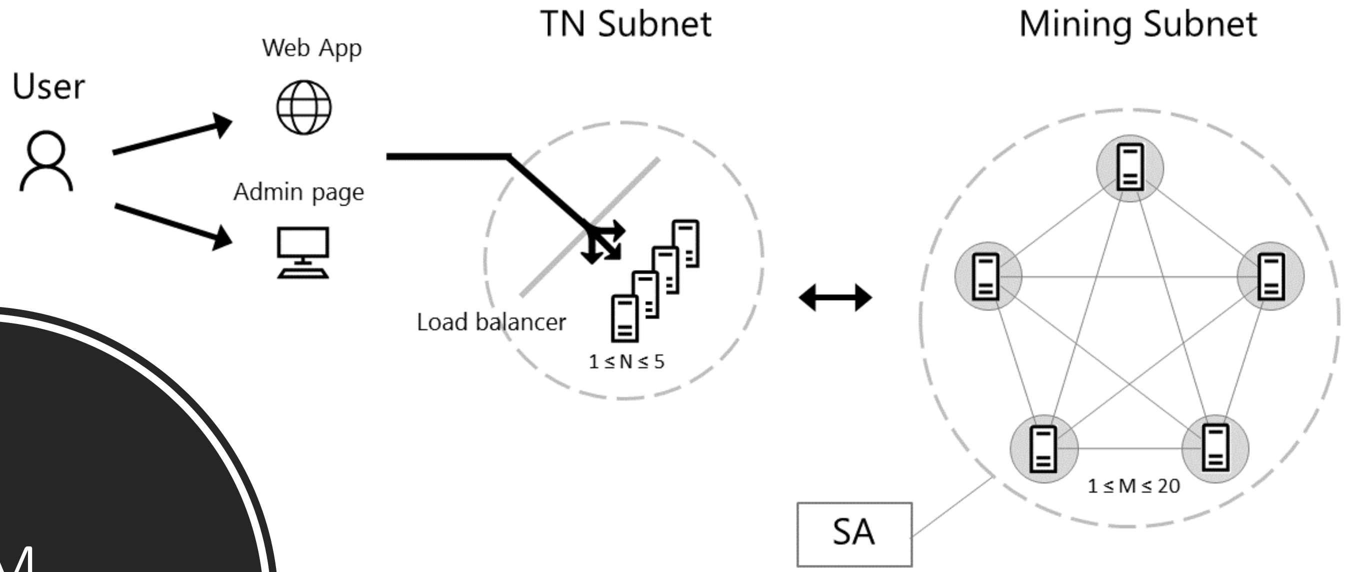
Post- ARM Template

To set up a mock consortium network in Azure today:

1. Review public Ethereum network **documentation**
2. ~~Determine topology for a consortium network~~
3. ~~Map topology to Azure resources (VMs, St~~ ~~.)~~
4. ~~Write ARM template or script OR~~
5. ~~Configure Ethereum cli~~ ~~scripts to support private network (peerin~~ ~~odes, etc.)~~
6. ~~Config~~ ~~protocol properties (genesis block, max peers, et~~
7. ~~Set~~ ~~hereum accounts and allocate ether~~
8. ~~Trial and error to make above steps work~~
9. Integrate with other **Azure services**, such as AAD and Key Vault
10. ~~Test template~~

5 minutes, 8 user parameters

ARM Template

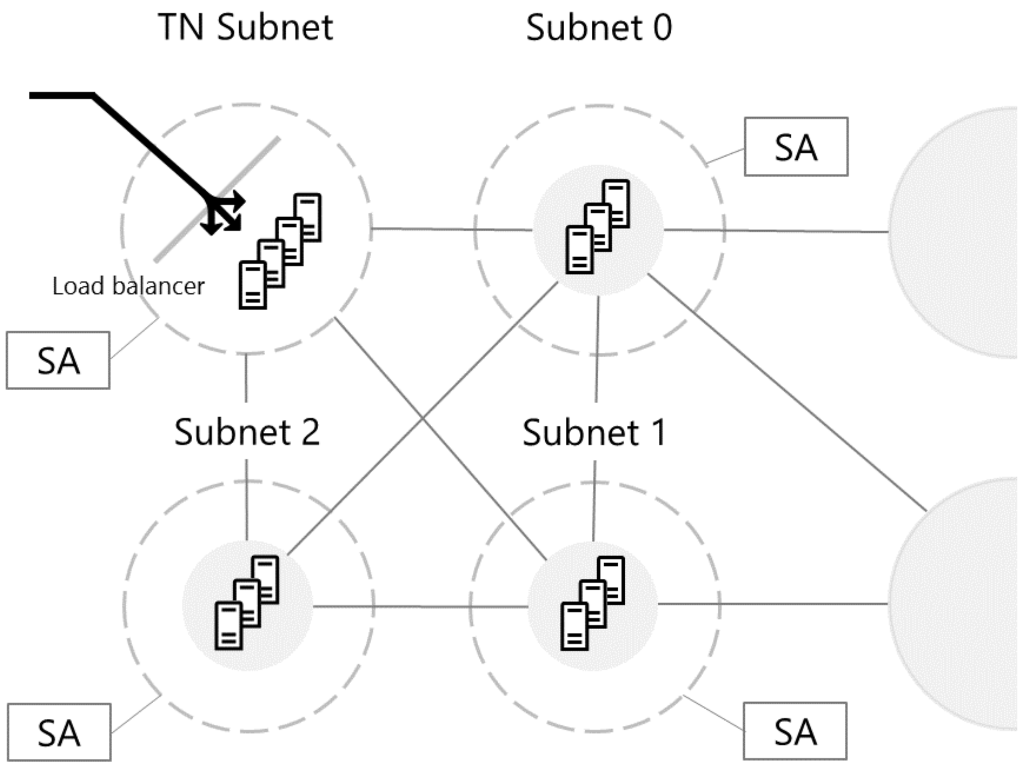


Simple N-Node deployment

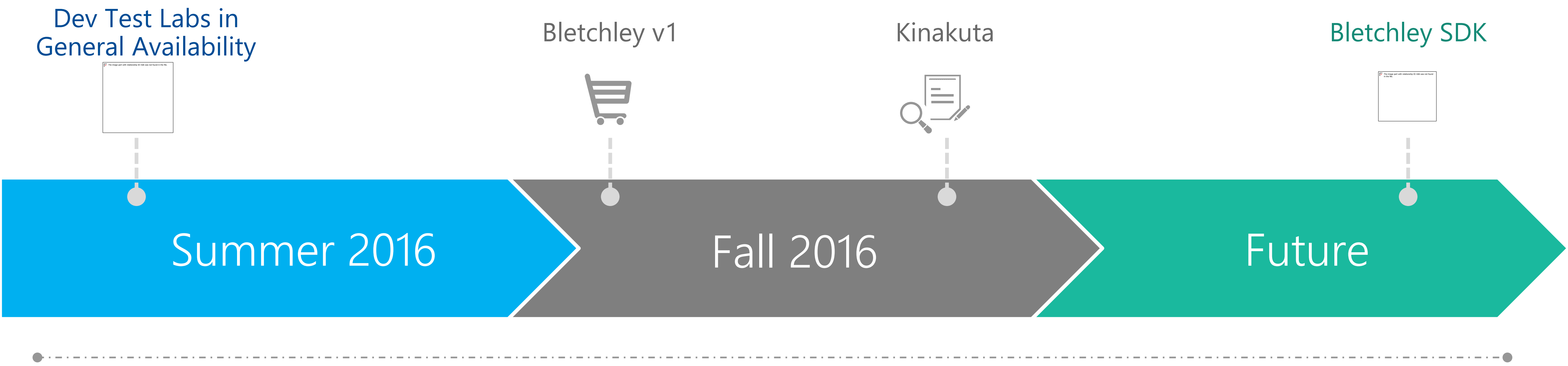
- 1 to 5 transaction nodes
- 1 to 20 mining nodes
- 2 subnets
- 2 storage accounts

Mock Consortium Network

- 1 to 5 transaction nodes
- 1 to 100 mining nodes
- 2 to 6 subnets
- 2 to 6 storage accounts



Microsoft BaaS | Roadmap



Try Today

- *http://azure.com/blockchain*
- *https://azure.microsoft.com/en-us/documentation/templates/*
- *Dev/Test BaaS Labs: https://github.com/marleyg/MSFTLabs/tree/master/DevTestBaaS*
- *43 different partners available today*

For Updates

- *https://azure.microsoft.com/en-us/blog/author/marleyg/*

Microsoft