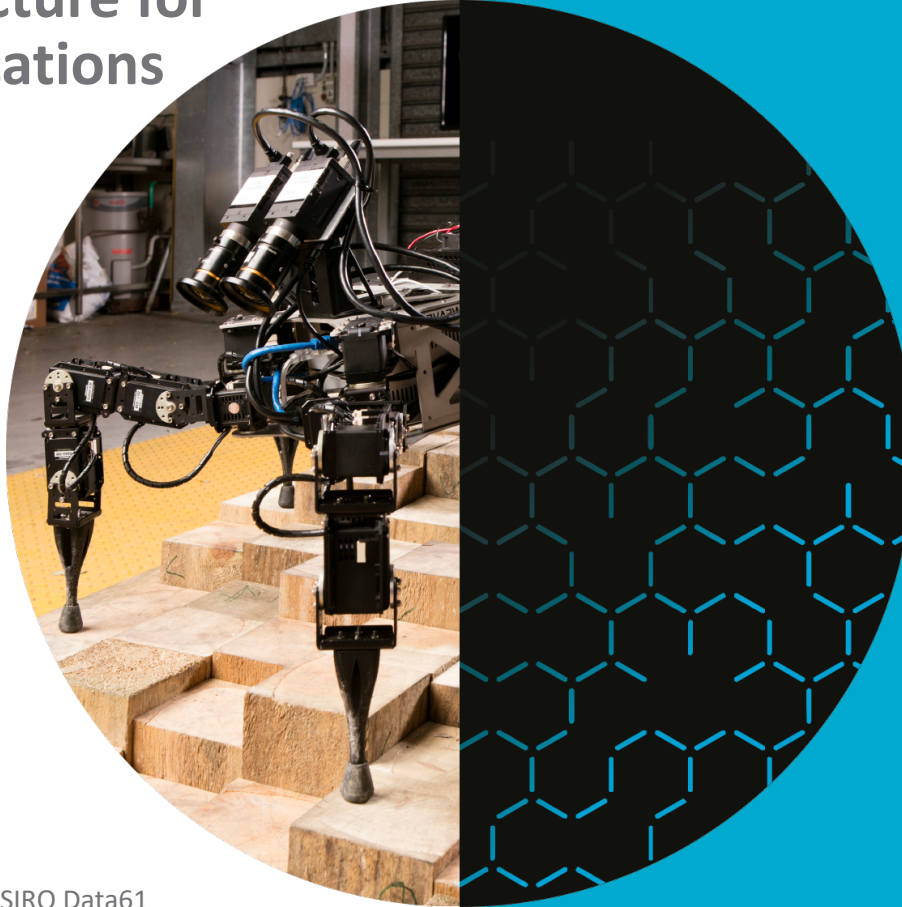


COMP6452

Software Architecture for Blockchain Applications



Cost

Helen Paik

| Senior Lecturer @ CSE, UNSW

| Visiting Researcher @ AAP team, CSIRO Data61

h.paik@unsw.edu.au

Australia's National Science Agency

Outline

- On-chain Data Cost
- Smart Contract Cost
- Cost Models
- Using and Evaluating the Cost Model



On-chain Data Cost

Cost is important

- Monetary cost is as important for blockchain technologies as they are for conventional technologies
- Blockchain systems have different cost models
- Cost for storing too much data on-chain can explode quickly
- Blockchains enable decentralized trust, but bring tradeoffs against execution cost and latency
- Cost may be inversely proportional to throughput on blockchain
 - E.g., on Ethereum: higher gas consumption of a Tx means lower Tx throughput per block / time (if all else is static)
 - Thus, cost is also relevant for non-public blockchains

Arbitrary Bytes in Unspendable Bitcoin Tx

Coinbase Transaction in Block

- Coinbase transaction mints new coins received by the miner who generates the block
- *Coinbase* parameter can contain arbitrary data
- Only the miner has access to this parameter

nSequence of a transaction

- Distinguish transactions from other Bitcoin transactions
- Presenting assets other than BTCs
- Every participants with the permission to submit transaction can set the value

Fake account address


- Sending a small amount of coins to the fake account
- The coin is lost forever
- Use 1-of-n multi-sig transaction
- Minimum amount of funds to avoid denial of service attack

Conditional statements

- *OP_IF*, *OP_ELSE*, *OP_ENDIF*
- Clauses cannot be reached under any condition
- Extra overhead

OP_RETURN

```
"vout" : [
  {
    "value" : 0.00000000,
    "n" : 0,
    "scriptPubKey" : {
      "asm" : "OP_RETURN 636861726c6579206c6f766573206865696469",
      "hex" : "6a13636861726c6579206c6f766573206865696469",
      "type" : "nulldata"
    }
  }
]
```



- Official way to embed arbitrary data in a Bitcoin transaction
 - Returns immediately with an error
 - Included data is not interpreted as a script
 - Default Bitcoin client only relayed *OP_RETURN* transactions up to 80 bytes
 - Reduced to 40 bytes in 2014
- Storing 80 bytes of arbitrary data on the Bitcoin costs roughly US\$0.459
 - Assuming a typical Bitcoin transaction with one input and one output (220 bytes)
 - The default transaction fee rate is 2×10^{-4} BTC/KB
- It is debatable whether Bitcoin should be used to record arbitrary data.

Exchange rates of US\$7650 / BTC from 2 August 2018 (https://poloniex.com/exchange#usdt_btc)

Storing Data on Ethereum Transaction

80 bytes on Bitcoin
costs US\$0.459

- Theoretically allows storing arbitrary data of any size
- Storing 80 bytes of arbitrary data on Ethereum costs roughly US\$0.22
 - Every transaction has a fixed cost of 21,000 gas
 - Gas is the internal pricing for executing a transaction of storing data
 - Every non-zero byte of data costs additional 68 gas
 - Total cost of storing 80 bytes via transaction is 26,440 gas
 - assuming all bytes are non-zero

Exchange rate of US\$420 / ETH from 2 August 2018 (https://poloniex.com/exchange#usdt_eth)
gas price of 2×10^{-9} ETH (2 Gwei) on Ethereum

Storing Data in Smart Contract

Storing 32 bytes of data
(simple types of Solidity are
32 bytes)

- Storing data as a variable in a smart contract
 - Cost is based on the number of *SSTORE* operations
 - 1 *SSTORE* operation that changes the data from zero to non-zero (20,000 gas)
 - Transaction as the carrier costs a base 21,000 gas
 - Data payload costs extra gas
 - Function signature and the actual data
 - Cost for creating the smart contract depending on its complexity
 - Total cost is $> \text{US\$}0.036$ ($20,000 + 21,000 + 32 \times 68$ gas)
 - Subsequent transactions to **update** data costs 5,000 gas (keeping the data as non-zero)
 - Cost of subsequent transactions is $\sim \text{US\$}0.024$ ($5000 + 21,000 + 32 \times 68$ gas)
 - Less flexible due to the constraints of Solidity on the value types and length
- Storing data as a log event in a smart contract
 - 1 log topic costs 375 gas
 - Every byte of data costs an extra 8 gas
 - Transaction as the carrier costs a base 21,000 gas
 - Total cost is $\sim \text{US\$}0.018$ ($21,000 + 375 + 32 \times 8$ gas)

Smart Contract Cost

Smart Contract Cost

- Cost charged on transactions in relation to their complexity
 - Base cost for any transaction (21, 000 gas)
 - Variable components
 - Data attachments
 - Contract execution is charged per *bytecode* instruction
 - Additional cost for contract deployment
- *Gas*
 - All cost follows a fixed pricing table specified in the unit *gas*
 - Official: [Yellow Paper](#) (see Appendix G)
 - Gas cost is converted to Ether
 - User-defined *gas price* factor
 - How much Ether-per-gas is the transaction creator willing to pay
 - Default value is the current market rate, an average over previously included transactions (but do not rely on it without testing)

Gas Limit

- **Block gas limit**
 - Sum of gas used by the set of transactions included in a given block cannot exceed this limit
 - Set by the miners
 - Defined in terms of gas usage
 - Cannot be influenced by variations the user has power over
 - For example, underbidding the market price
 - Making it a limit of complexity for new blocks
 - An upper bound to throughput scalability
 - Cost of transactions vary
 - Non-trivial to understand how the bound relates to transaction throughput
- Current prices & inclusion speeds: <https://ethgasstation.info/>

Cost Model and Use

Cost Modelling & Estimation

Question : What is the cost of distrust?

i.e., how much more expensive is using Ethereum over public cloud

To cost this out, we need operational costs of all components of an application ... to do this we will:

- present some cost equations to calculate different operation components in blockchains and cloud environments ...
- use the execution of an instance of a business process model (the bulk buyer scenario) as sample application to drive the cost calculations
- compare the costs of running a business process on blockchain vs. cloud

Two costing options: blockchain vs. cloud

- **Cost Model of Blockchain Infrastructure**
 - **Using Ethereum**
 - It is a representative Blockchain platform
 - It provides Turing complete language to implement business logic
- **Cost Model of Cloud Infrastructure**
 - **Using Amazon Web Services (Amazon SWF – Simple WorkFlow)**
 - It is dedicated to process execution
 - Implements commonly-used workflow patterns and messaging patterns
 - AWS is a leading commercial cloud computing provider

Ethereum Transaction 1/2

- 3 types of transactions
 - Financial transfer, message call and contract creation
 - Basic elements: `from`, `to`, `gasLimit`, `value` and `data`
- Financial transfer transaction
 - `From/to` sender/recipient of the transaction
 - `Value` the amount transferred
 - `Data` (optional) data in arbitrary form
 - E.g. JSON, XML, pictures, hash values
 - Fee for a transaction covers the cost for storing the data permanently

Ethereum Transaction 2/2

- Message call transaction
 - Invoke a function of a contract
 - From/to sender/recipient of the transaction
 - Value (optional)
 - Data the method to be invoked and the parameters
 - gasLimit maximum gas can be used in this transaction
 - Gas is paid for each executed bytecode instruction
- Contract creation transaction
 - to NULL
 - Data the contract bytecode
 - Value (optional)

Cost of executing
business process

Cost of deploying
business process

Contract Creation Cost 1/2

- Contract creation transaction
 - Data compiled bytecode

$$C_{pload} = \text{payload (in bytes)} \times C_{gas/byte}$$

- Permanent storage of this data incurs cost

- Value (optional Ether transfer)
 - “Endowment” upon initialization
- Ethereum address is assigned to it
 - Calculated with a deterministic function depending only on the creator’s Ethereum account
- Cost for contract creation:

$$C_{create} = C_{tx} + C_{addr} + C_{pload} + C_{fndef}$$

- C_{tx} : 21,000 gas base cost for transaction itself
- C_{addr} : 32,000 gas for allocating address
- C_{fndef} : consumed by the opcodes in the function definition

- Cost of payload for contract bytecode is 200 gas per byte
- Cost of payload for data in a financial transaction/message call is 68 gas per non-zero byte and 4 per zero byte

Contract Creation Cost 2/2

- Contract can be created by another contract
- Cheaper Without C_{tx}

$$C_{create}' = C_{addr} + C_{pload} + C_{fndef}$$

Contract Execution Cost

- Function call cost

$$C_{execute} = C_{tx} + C_{pload} + C_{fnexe}$$

- C_{tx} : 21,000 base gas for transaction itself
- C_{pload} : cost of data payload
- C_{fnexe} : consumed by the opcodes executed **during** the function invocation

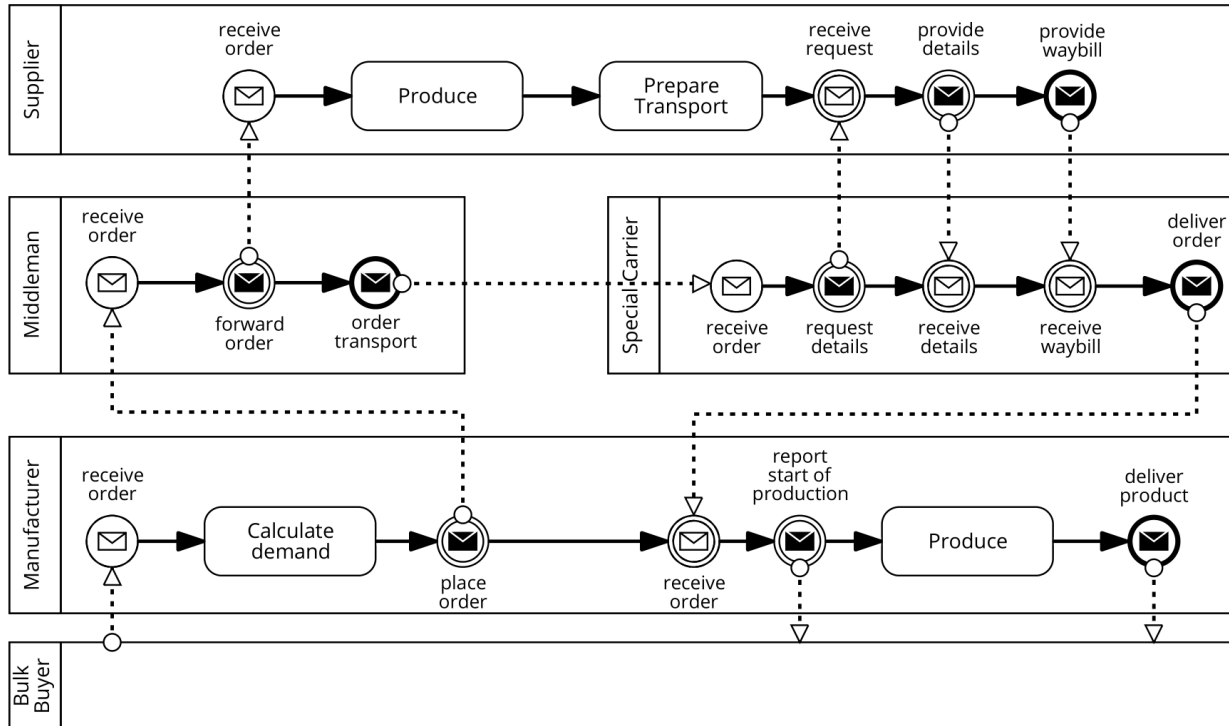
Gas Cost → Ether → Another Currency

$$C_{\$} = C_{Gas} \times \text{gasPrice} \times 10^{-18} \times \text{EXC}_{ETH2CUR}$$

- $C_{\$}$: cost in \$
- C_{Gas} : cost in gas
- Gas price in *wei*
 - *then convert to Ether* ($1 \text{ wei} = 10^{-18} \text{ Ether}$)
- $\text{EXC}_{ETH2CUR}$: Exchange Rate from Ether to Currency
 - See e.g.
 - <https://coinmarketcap.com/currencies/ethereum/>
 - <https://fx-rate.net/ETH/USD/>

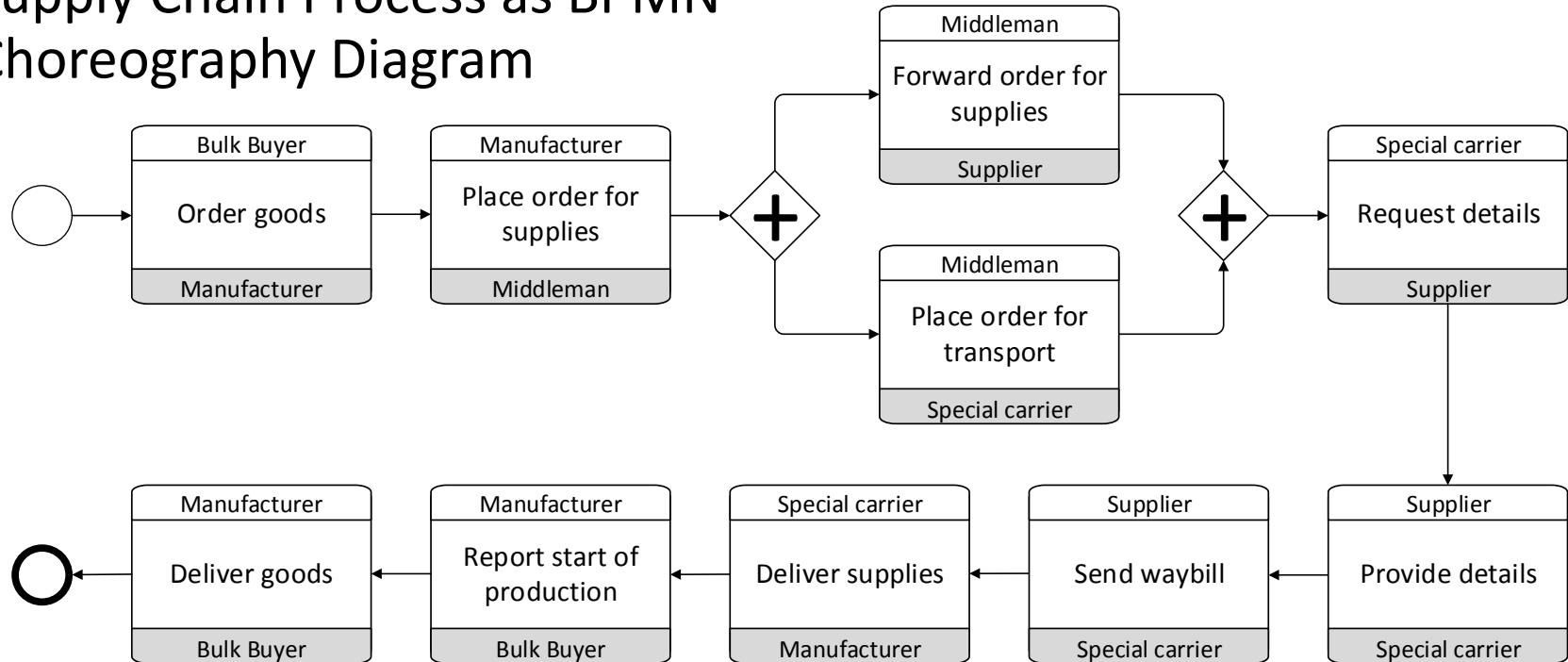
Cost of Interaction Component 1/5

- Supply Chain Process as BPMN Orchestration Diagram



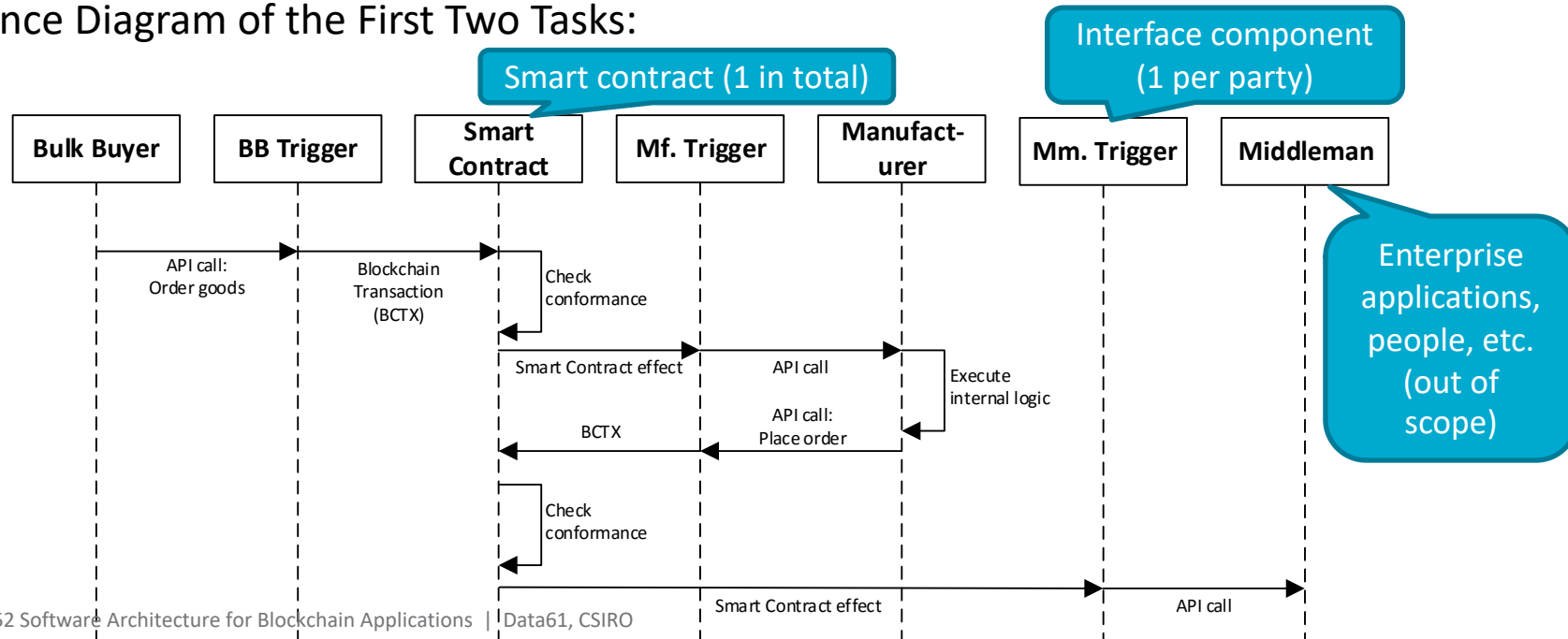
Cost of Interaction Components 2/5

- Supply Chain Process as BPMN Choreography Diagram



Cost of Interaction Components 3/5

- Interaction between internal process implementation, triggers, and process instance smart contract
- Sequence Diagram of the First Two Tasks:



Cost of Interaction Component 4/5



- Assuming the interface component is running on AWS

$$C_{comp} = EC2_{price}(ec2_t) \times time$$

- $EC2_t$: the set of all available VM types in AWS; $ec2_t \in EC2_t$ is the chosen type
- Capacity (Throughput) of VM
 - $TP_{bc} : EC2_t \rightarrow R$
- Determine VM type needed based on workload WL_{bc}
 - $f_{bc} : (TP_{bc}, WL_{bc}) \rightarrow EC2_t$
- Cost of running a VM of this type per billing time unit (BTU)
 - $EC2_{price} : EC2_t \rightarrow R$

Cost of Interaction Component 5/5



- Interface component operates a full node
 - Synchronize the blockchain if the VM is not constantly online
 - Required duration includes the time of synchronization
- Ethereum clients allows fast synchronization (“fast” flag)
 - Downloading transaction receipts instead of the full set of known blocks
 - Shows that the transactions happened but
 - Does NOT show the results of the smart contract function execution
 - Less evidence for integrity
 - Can only be done when downloading the blockchain from scratch
 - Takes on the order of hours to days for public Ethereum blockchain

Two costing options: blockchain vs. cloud

- **Cost Model of Blockchain Infrastructure**
 - **Using Ethereum**
 - It is a representative Blockchain platform
 - It provides Turing complete language to implement business logic
- **Cost Model of Cloud Infrastructure**
 - **Using Amazon Web Services (Amazon SWF – Simple WorkFlow)**
 - It is dedicated to process execution
 - Implements commonly-used workflow patterns and messaging patterns
 - AWS is a leading commercial cloud computing provider

Amazon SWF

- Simple Workflow Service (SWF) provided by AWS
- Representative for cloud-based business process execution
- Clear mapping to the process model
- Tiered pricing model
 - More usage results in cheaper cost per unit

Workflow Executions

A workflow is a set of tasks executed in a certain order (sometimes with a set of conditional flows or loops). Each time that a workflow is executed, it is considered a distinct workflow execution. You pay for workflow executions when you start them (i.e. their first task becomes available for application hosts to execute) and for each 24-hour period until they are completed. The first 24 hours of workflow execution are free.

Start a Workflow Execution

Region: US East (Ohio) ↕

- \$0.00 for first 1,000 workflow executions
- \$0.0001 per workflow execution above the free tier

Data Transfer **

Region: US East (Ohio) ↕

Data Transferred	Pricing
Data Transfer IN	
All data transfer in	\$0.00 per GB
Data Transfer OUT***	
Up to 1 GB / Month	\$0.00 per GB
Next 9.999 TB / Month	\$0.09 per GB
Next 40 TB / Month	\$0.085 per GB
Next 100 TB / Month	\$0.07 per GB
Greater than 150 TB / Month	\$0.05 per GB

SWF Cost Model

- Main elements: workflow, actor, task, and signal
- Workflow: A collection of activities in a specified sequence
 - An instance of business process
- Actor: Play participant roles from the business process.
- Activity (task): Schedule a notification to the appropriate actors to proceed with the next activity
- Decision (task):
 - Determine whether the current state of execution conforms to the workflow
 - Determine which activity to execute next
- Signal: External triggered event to a currently executing workflow

Element Mapping

Business Process	Blockchain	Amazon SWF
Process instance	Instance of Smart Contract	Workflow
Conformance checking	Contract execution (Partial)	Decision task
Activity	Contract execution (Partial)	Activity task
Incoming message	Transaction	Signal
Outgoing message	Entry in contract event log	Notification

Base Cost of Workflow Instances

$$C_{wf} = \#wf \times SWF_{wf}$$

- $\#wf$: Number of instances
- SWF_{wf} : SWF cost of starting a workflow execution

Cost of Scheduling Tasks

$$C_{task} = (\#actTask + \#decTask) \times SWF_{task}$$

- *#actTask* : number of activity tasks
- *#decTask* : number of decision tasks
- SWF_{task} : price per task
- No. SWF activity tasks = No. activities in a process instance
- No. SWF decision tasks = No. activities in a process instance + 1
 - *Why number of activities* → SWF schedules a decision task every time it receives a signal (== completion of an activity)
 - *Why +1* → For each process instance, the initialization creates a new workflow (instance) and a decision task to instruct the workflow to wait for the first signal.

Cost of Signals

$$C_{sig} = \#signals \times SWF_{signal}$$

- *#signals* : number of signals
 - (can be observed from the number of activities in a business process instance)
- SWF_{signal} : price per signal

Cost of Data Retention and Transfer

$$C_{ret} = (execT + retT) \times SWF_{ret}$$

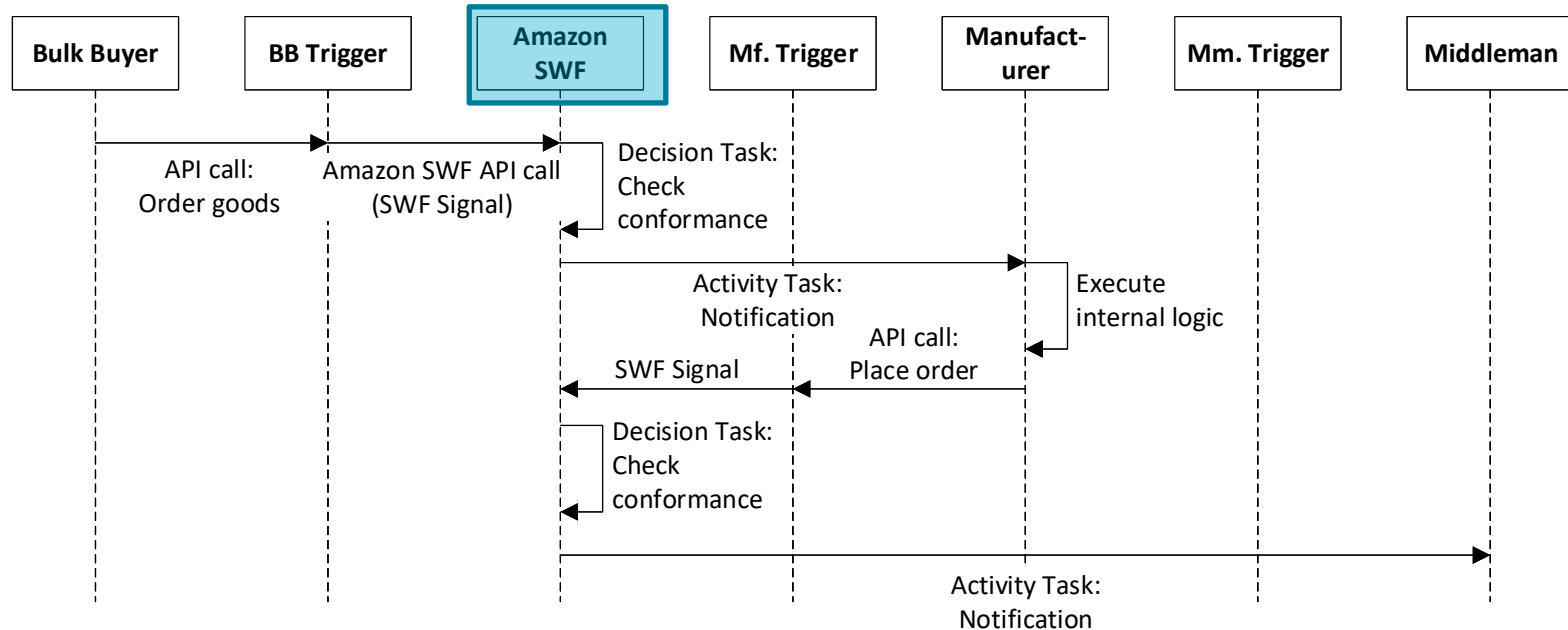
$$C_{dat} = payload \times SWF_{data}$$

- *retT*: user-specified duration for generated data being retained
 - Charged for storage per 24 hours
- *execT*: Workflow execution time
 - Charged per 24 hours at the same rate as data retention cost
- SWF_{ret} : SWF cost rate
- *Payload*: inwards and outwards data size
- SWF_{data} : price per data unit

Coordination Cost (on SWF)

$$C_{swf} = C_{wf} + C_{task} + C_{sig} + C_{ret} + C_{dat}$$

Cost of Interaction Component 1/2



- Task execution requires actor running a *Amazon SWF worker* module
 - On AWS EC2 or internal infrastructure
 - Execute both decision task and activity task

Cost of Interaction Component 2/2

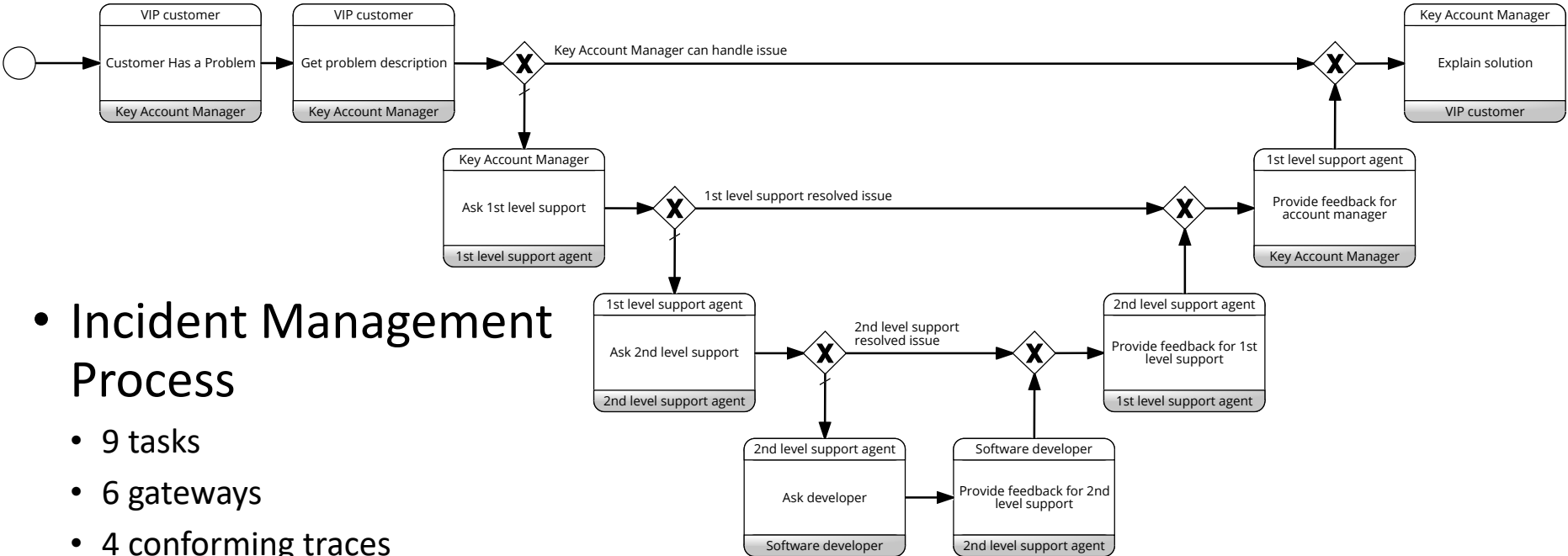
Cost of VMs

- For running Triggers and Amazon SWF workers
- Consider capacity(throughput) per VM type

$$TP_{swf} : EC2_t \rightarrow R$$
- Chosen VM type depends on capacity of VM types and the workload
 - The throughput values are different from the ones for blockchain triggers
$$f_{swf} : (TP_{swf}, WL_{swf}) \rightarrow EC2_t$$
- Cost of running the VMs for the time needed

$$C_{comp} = EC2_{price}(ec2_t) \times time$$
- Minimum requirement is one VM to host the trigger and worker
- Preferable setup is that each participant at least one VM
 - Host their own trigger and worker

Evaluating Cost Model



• Incident Management Process

- 9 tasks
- 6 gateways
- 4 conforming traces
- Cross-organizational
- First-level support outsourced

AWS VM Throughput Benchmark

VM Types	vCPU Specifications	Memory (GiB)
t2.small	1 Intel Xeon E5-2676 2.40 Ghz v3 w/ Turbo up to 3.3 Ghz	2
m3.medium	1 Intel Xeon E5-2670 2.50 GHz v2 (Ivy Bridge) Processors	3.75
m3.large	2 Intel Xeon E5-2670 2.50 GHz v2 (Ivy Bridge) Processors	7.5
m3.xlarge	4 Intel Xeon E5-2670 2.60 GHz v2 (Ivy Bridge) Processors	15

Decision limits

Decision	Bucket size	Refill rate / s
RequestCancelExternalWorkflowExecution	100	10
ScheduleActivityTask	500	100
SignalExternalWorkflowExecution	500	10
StartChildWorkflowExecution	500	2
StartTimer	1000	142

- AWS EC2 VM types and specification

- Throughput

Metrics	Blockchain			Amazon SWF	
	t2.small	m3.medium	m3.large	m3.medium (default)	m3.medium (incr. limit)
Transactions or Signals	13,580	7,336	20,104	73,871	152,404
Network In (MB)	102	114	128	138	168
Network Out (MB)	195	131	278	353	376
Duration (sec)	3,610	3,605	3,604	3,605	3,605
Average Tx/sec or Average signal/sec	3.8	2.0	5.6	20	42

Incident Management Blockchain Cost

- 32 process instances with a total 256 transactions
- Deployment of factory contract costs 0.032 Ether (One-time cost)
- Each run with data transformation costs 0.0347 Ether
- Total cost is approx. US\$1.34
 - Exchange rate is US\$420 / ETH
 - Gas price of 2 *Gwei*

Incident Management Amazon SWF Cost

- EC2 *t2.micro* VM for trigger and SWF task worker
- Process instances executed in sequence

- US\$0.92 for 1,000 process instances

- US\$0.000925 per instance
 - Data retention is 1 day
- US\$0.002745 per instance
 - Data retention is 365 days

- Cost breakdown



Element	elements in experiment	Unit cost (US\$)	Total cost (US\$)
Decision Task	15,000	0.000025	0.375
Activity Task	7,000	0.000025	0.175
Signal	7,000	0.000025	0.175
Workflow	1,000	0.0001	0.1
Retention (24h)	1,000	0.000005	0.005
Execution Time (24h)	1,000	0.000005	0.005
Data Transfer	1	0.09	0.09

Comparison

- Process instance on blockchain is three orders of magnitude higher than on Amazon SWF
 - Excluding the one-time factory contract deployment
- Blockchain stores the result in perpetuity
 - As long as the blockchain is in existence
- Ongoing cost for data storage on Amazon SWF
 - Store for 243,863 days (approx. 668 years) to reach break-even (with a rate of US\$420 / ETH)

Volatility of Cryptocurrency

- Sensitive to the volatility of the exchange rate

Costs	Ethereum (in Ether)	Exchange Rate (in US\$)				
		0.10	1.00	10.00	100.00	1,000.00
Incident Management (contract deployment)	0.0032	0.00032	0.0032	0.032	0.320	3.20
Incident Management (per process instance)	0.00347	0.000347	0.00347	0.0347	0.347	3.47

(+n) signify order of magnitude higher more expensive on Blockchain.

- Comparison

- Exchange rate
- Retention rate
 - (how long the data is stored)

Costs	SWF cost (in US\$)	SWF vs Blockchain cost comparison in ratio with different exchange rates (ratio < 1 means Blockchain is cheaper)					Break-even rate (in US\$)
		\$0.10	\$1.00	\$10.00	\$100.00	\$1,000.00	
Incident (24 hours)	0.000925	0.375 (0)	3.751 (+1)	37.51 (+2)	375.14 (+3)	3,751.35 (+4)	0.27
Incident (99 years)	0.181595	0.002 (0)	0.019 (0)	0.191 (0)	1.91 (0)	19.11 (0)	52.33

99 years (long term)



Why Blockchain

- Blockchain provides trustworthy storage and execution environment
 - No trust in any single third-party
- Conventionally participants need to jointly agree on a mutually-trusted third party
 - E.g. AWS (for confidentiality and truthful execution)
 - The party controlling the Amazon SWF account
- Public blockchain supports payment and escrow
 - Sending cryptocurrency with existing messages would not incur additional cost
 - Due to a flat fee structure
 - Offset the premium cost of distrust
 - Commercial escrow service charge 0.5% to 3.25%
 - Lower the cost of process executions involving monetary transaction
 - Only possible if acceptable crypto-coin can be established or used

Co-opetition

- Organizations cooperate for cases to achieve business goals
- Compete in other cases

Cost vs. Maintainability

- Deployment methods impact cost and non-functional properties
 - (1) One smart contract with two functions
 - (2) Two smaller contracts, each implementing one function
 - One contract acts as an entry point
- The first has lower deployment cost
 - For (2), one needs to pay C_{tx} and C_{adr} twice
 - The payload of contracts in (2) is higher, as there are header bytes in the payload
- (1) is cheaper but is not as maintainable as (2)
 - One function needs to be modified
 - (1): updated contract needs to be redeployed as a whole
 - (2): only one contract is redeployed
 - (1): the triggers need to be updated with the new address
 - (2): might be avoided (if the entry point hasn't changed)

Cost vs. Scalability of Triggers

- Additional resources are needed to accommodate increasing workload
 - Vertical scaling (bigger VM)
 - Horizontal scaling (more VMs)
- Blockchain nodes can scale vertically
 - Horizontal scaling has complication
 - Easy to add additional VMs into the network
 - Using one account from multiple VMs may lead to double-spending
 - Using different accounts on different VMs
 - Maintainability issues and increase storage costs
- SWF can scale both horizontally and vertically
 - Vertical scaling by choosing larger VM
 - Horizontal scaling by adding new VMs
 - SWF then acts as load balancer

Summary

- Cost of basic compute and storage on public blockchain have different cost structure than conventional cloud
 - Orders of magnitude more expensive
- Public Ethereum and Amazon SWF are compared using business process execution
 - Construct and benchmark cost model for both infrastructures
 - Cost on public Ethereum blockchain is three orders of magnitude higher than on Amazon SWF
 - But there may be good reason to still consider blockchains ...
- Cost model incorporates exchange rate is important
 - Given the high volatility of the exchange rate
- Cost is often in tradeoff with other non-functional properties
 - Maintainability and Scalability



Thank You

Helen Paik

| Senior Lecturer @ CSE, UNSW
| Visiting Researcher @ AAP team, CSIRO Data61
| h.paik@unsw.edu.au