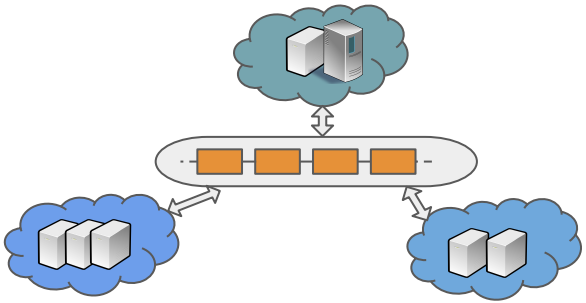
Towards a Democratic Federation for Infrastructure Service Provisioning



Bishakh Chandra Ghosh (ghoshbishakh@iitkgp.ac.in), Sourav Kanti Addya, Anurag Satpathy, Soumya K Ghosh and Sandip Chakraborty

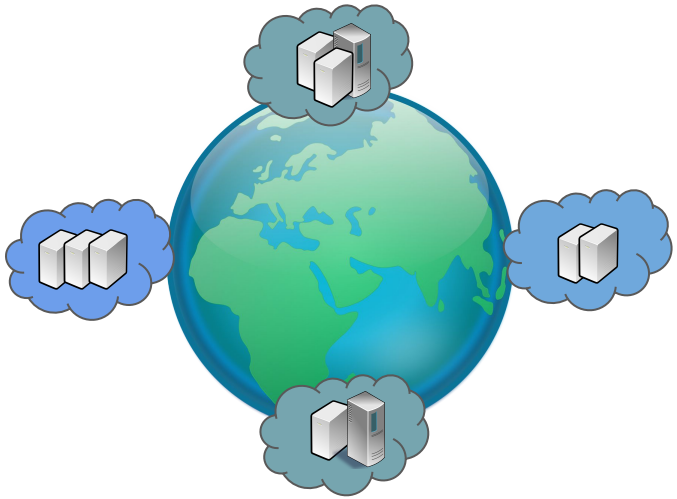
Department of Computer Science & Engineering

Indian Institute of Technology Kharagpur

CNeRG

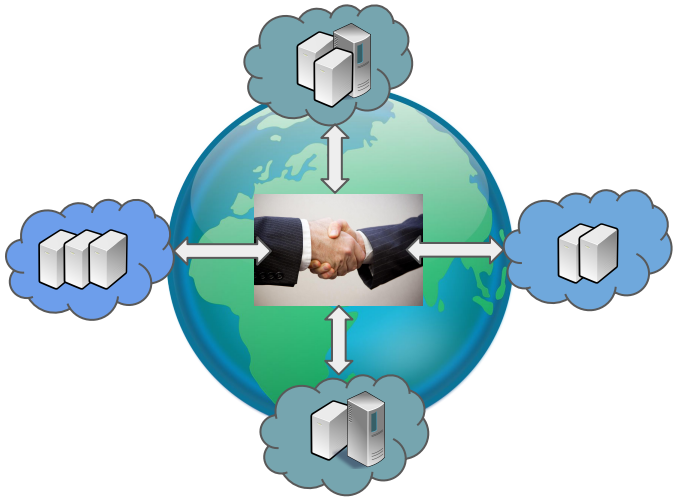
Cloud Federations

Collaboration among different **Cloud Service Providers** (CSPs), whereby they agree to mutually share their own resources for their overall benefit.



Cloud Federations

Collaboration among different **Cloud Service Providers** (CSPs), whereby they agree to mutually share their own resources for their overall benefit.



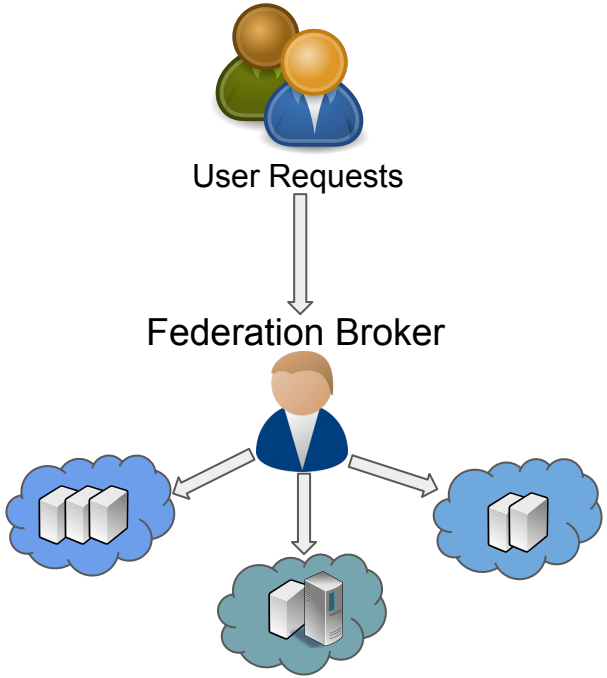
Motivation for cloud federation

- **Sharing** of computing resources.
- **Aggregation of unused resources** from different service providers.
- Bringing services closer to customers by maximizing the **geographical dispersion**.
- **Tackling data protection laws** that requires data to be stored within country’s boundary.

Existing cloud federations

Mostly **Centralized** Approach:

1. **Centralized cloud broker**



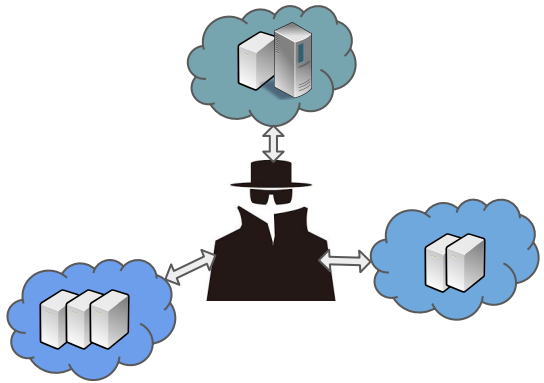
Existing cloud federations

Mostly **Centralized** Approach:

2. Centralized cloud exchange

Limitations of existing federations:

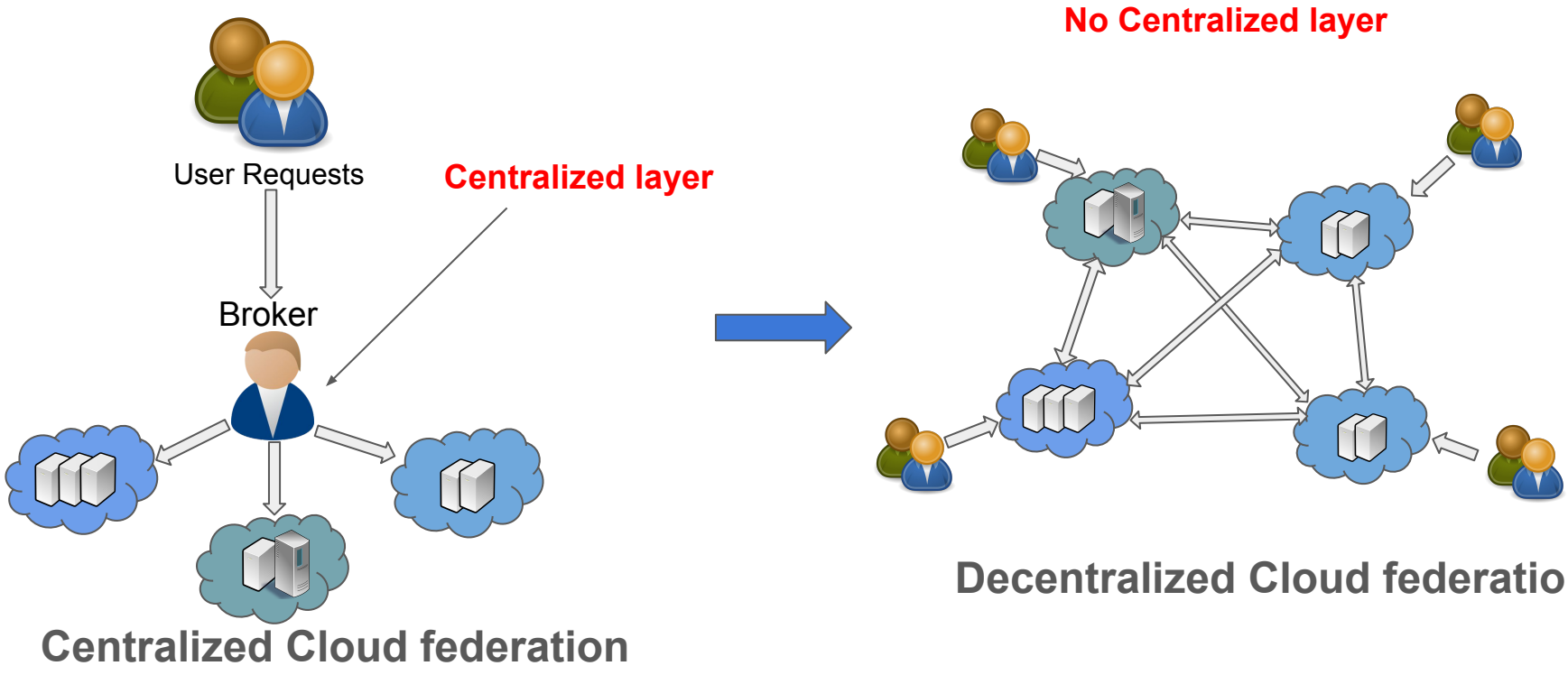
1. Profit sharing with central broker
2. Biasness of broker towards certain service providers
3. Price manipulation (*Broker can be malicious*)
4. Unfair dispute resolution
5. Central point of failure



Objective

Remove the central broker **and design a transparent distributed system for cloud federation.**

Centralized to Decentralized



Some related works

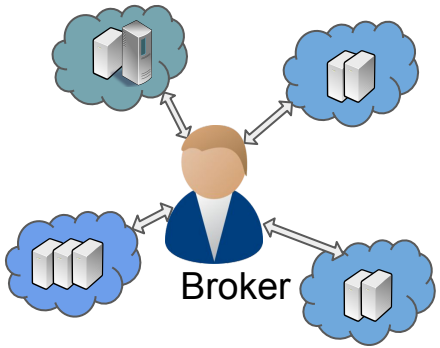
S. Kirkman	Data movement policy framework using smart contracts[1]	Not concerned with multiple CSPs
Sukhodolskiy et al.	Blockchain-based access control system for cloud storage[2]	
Zhou et al.	Blockchain based witness model for SLA monitoring[3]	
Margheri et al.	Distributed infrastructure for democratic cloud federations[4]	Public Blockchain

[1] S. Kirkman, "A data movement policy framework for improving trust in the cloud using smart contracts and blockchains," in IEEE IC2E, 2018, pp. 270–273.
[2] I. Sukhodolskiy and S. Zapechnikov, "A blockchain-based access control system for cloud storage," in IEEE EIConRus, 2018.
[3] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in IEEE INFOCOM, 2019.
[4] A. Margheri, M. S. Ferdous, M. Yang, and V. Sassone, "A distributed infrastructure for democratic cloud federations," in IEEE CLOUD, 2017, pp. 688–691.

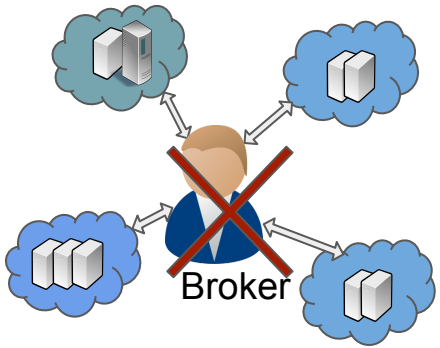
Challenges

- A decentralized platform for exchange of infrastructure resources (VM) must be developed.
- The system must allow coordination between service providers while enforcing FLA, without the help of any broker.
- Cloud functions such as VM Placement and VM Migration needs to be coordinated over the decentralized architecture.
- Fair ordering of transactions must be ensured

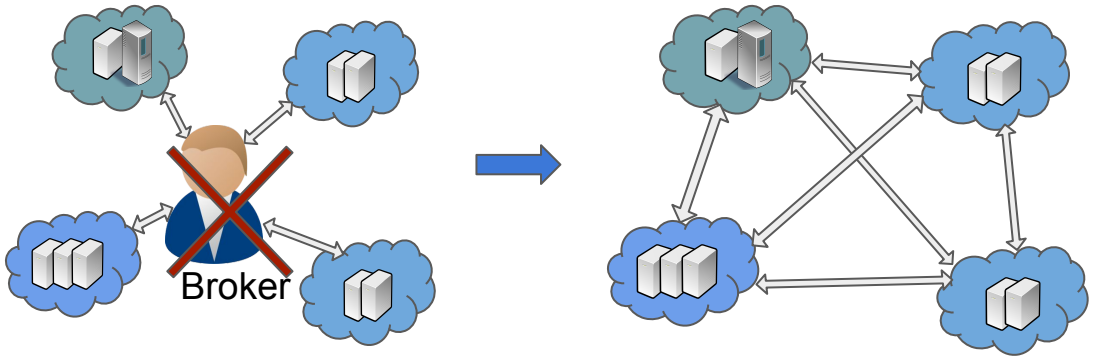
Proposed Architecture



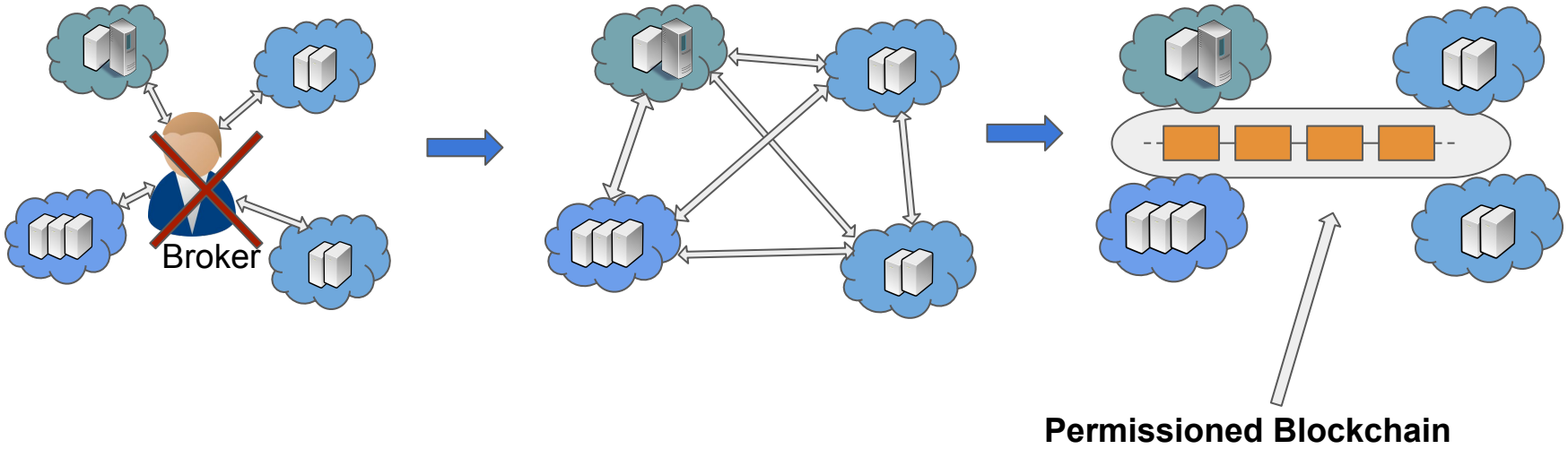
Proposed Architecture



Proposed Architecture



Proposed Architecture



Proposed Architecture

Permissioned blockchain based **decentralized exchange** for **democratic** cloud federations: ***CloudChain***

Proposed Architecture

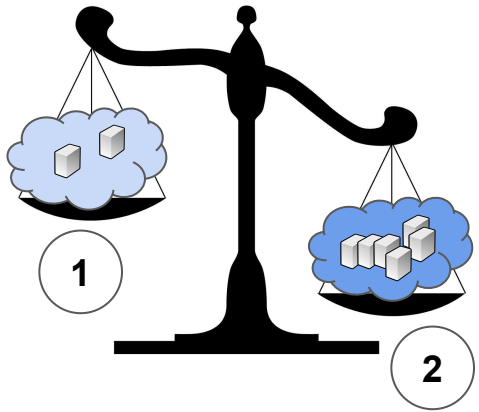
For simplicity we assume that the federation contains two types of service providers namely,

1. Demanding service providers:

   Suffer from **resource limitations** and require other members of the federation to create instances for them at peak loads.

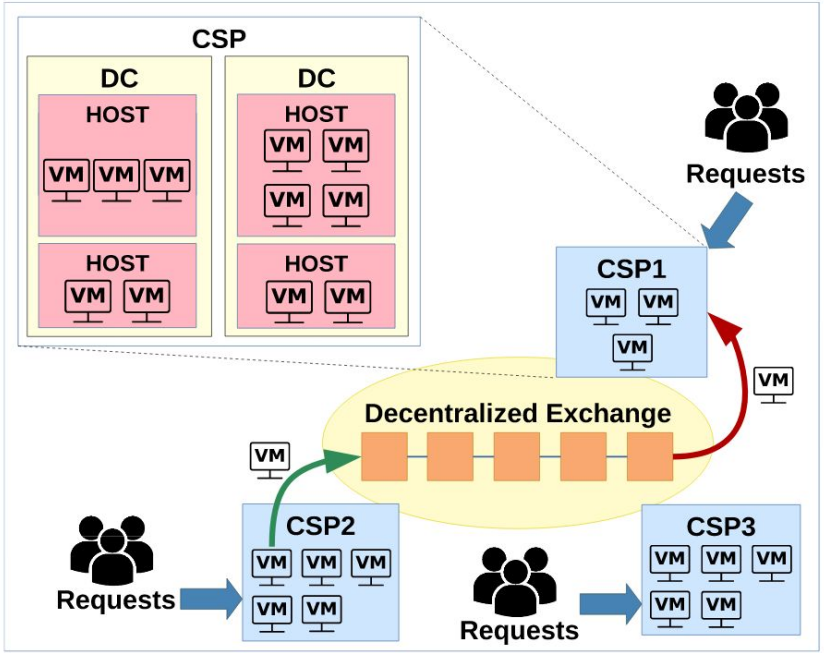
2. Supplying service providers:

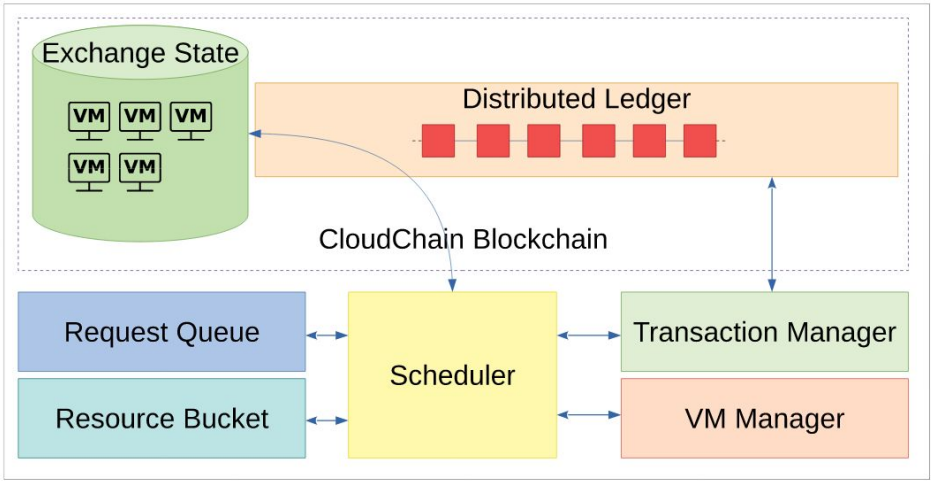
   Having **abundant resources** which goes unused.



Proposed Architecture

CloudChain





Components of CloudChain

CloudChain model overview

Components

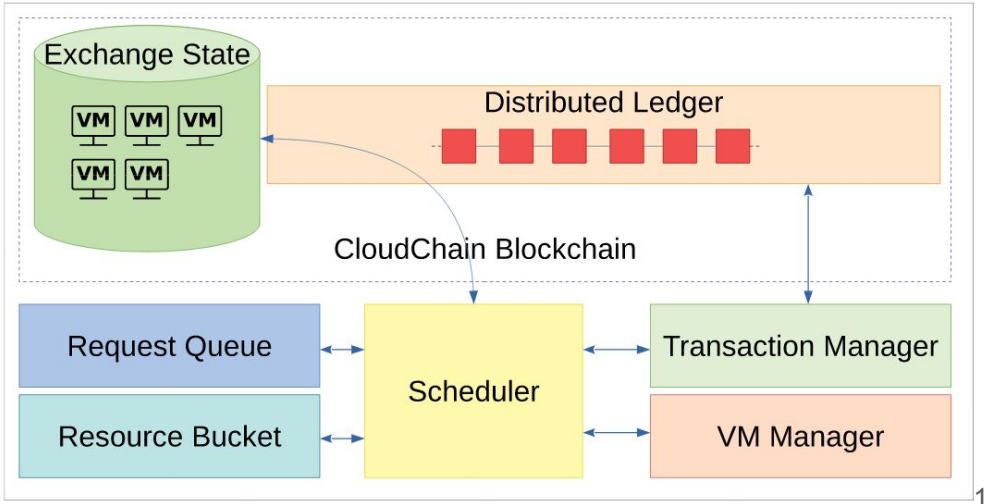
1) *CloudChain* Blockchain: Distributed Ledger & Exchange State

2) Request Queue and Resource Bucket

3) Scheduler

4) Transaction Manager

5) VM Manager



Components

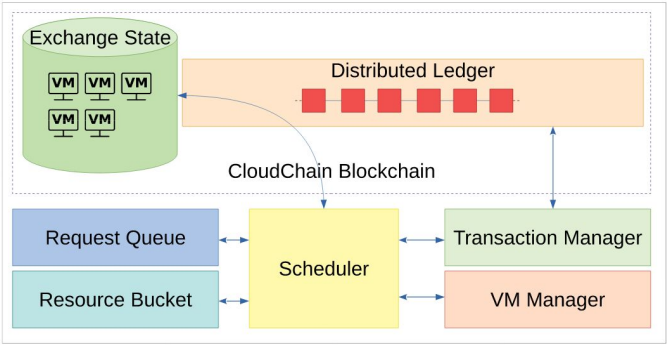
Request Queue (*ReQ)*: queue of incoming multi-tier web application requests

Resource Bucket:

Bucket of available resources, which may include both local resources and exchange resources.

1) Local resource bucket (*ResBlocal*)

2) Exchange resource bucket (*ResBexchange*)

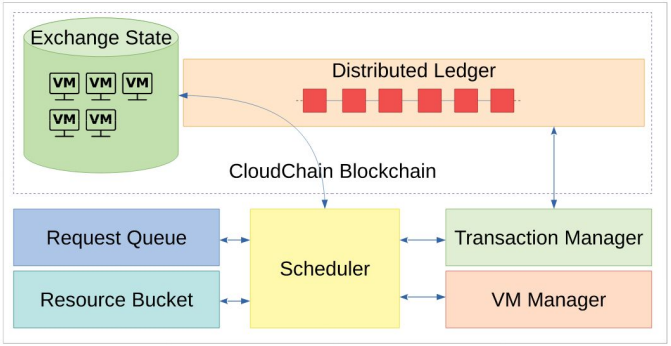


Components

Scheduler: Coordinates all the components of *CloudChain*

Transaction Manager: An interface to the *CloudChain* Blockchain.

VM Manager: Manages creation, deletion and access to VMs.



***CloudChain* Blockchain** serves as an information registry that maintains the current state of available resources and demand patterns. Thus it acts as a common marketplace where different CSPs can offer their unused or excess resources for outsourcing, and rent resources from other CSPs when required.

Exchange State

Offerings			
ID	Supplying SP	Specs	Price/BTU
o_1	Cloud 1	2 Cores, 4GB, UK	15$/Month
o_2	Cloud 2	2 Cores, 8GB, US	18$/Month
o_3	Cloud 1	2 Cores, 2GB, UK	0.16$/Day

Requests			
ID	Demanding SP	Offering Id	Duration
r_1	Cloud 2	o_1	6 Months
r_2	Cloud 3	o_2	12 Months

Associations			
ID	Demanding SP	Offering Id	Duration
a_1	Cloud 3	o_5	3 Months
a_2	Cloud 5	o_4	7 Days

CloudChain Blockchain

The high level **operations** that the CSPs can perform on the exchange are:

1) **Offer** a new resource

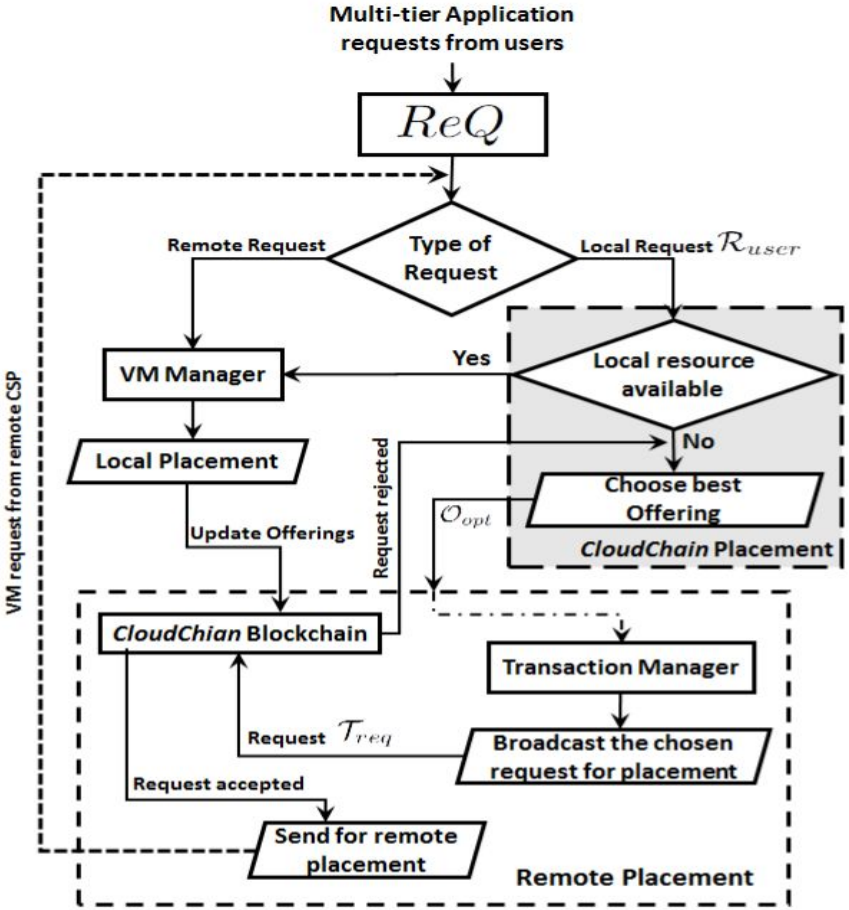
2) **Modify** an existing offering

3) **Query** for available resources offerings

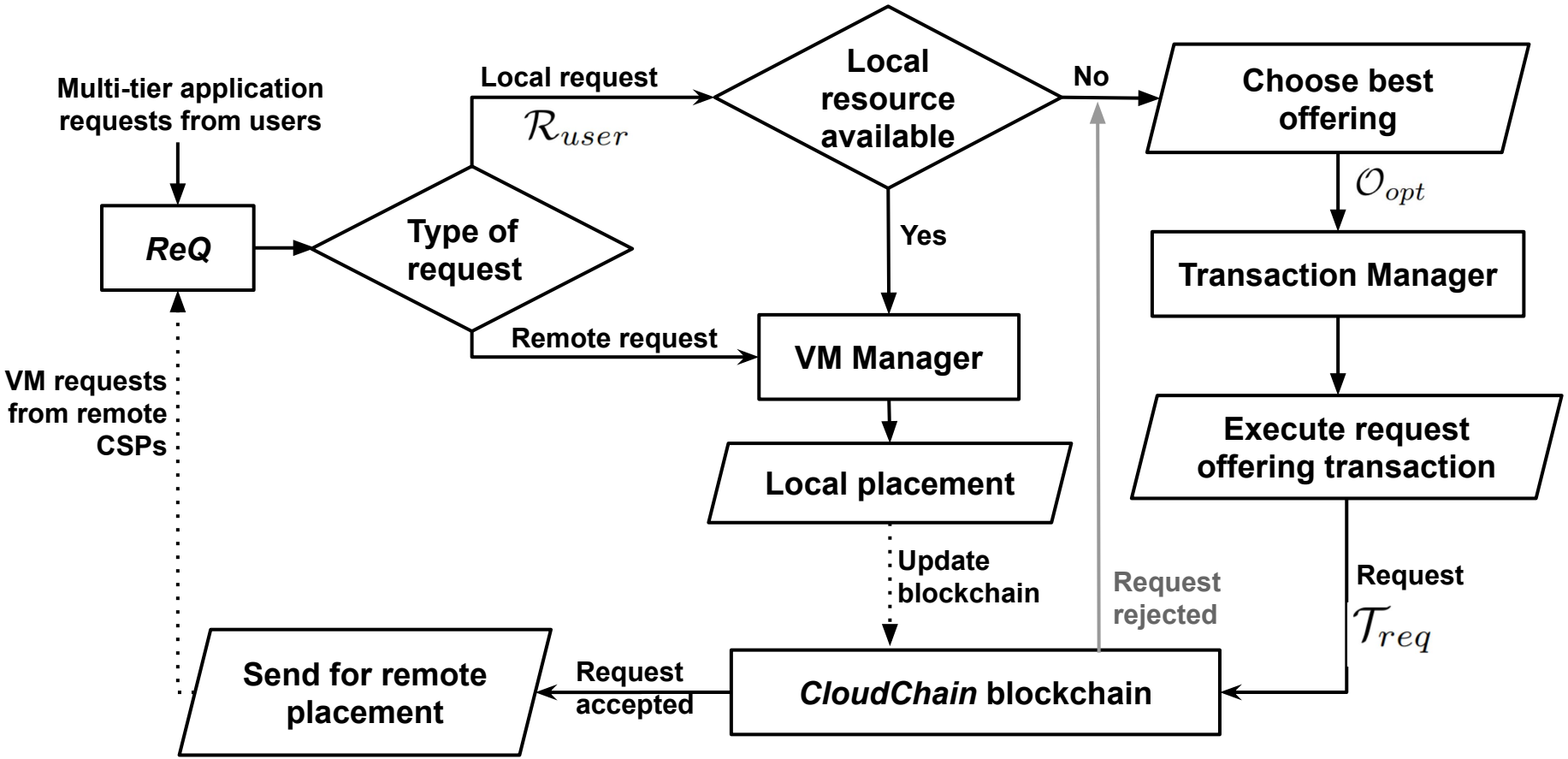
4) **Request** to rent a resource

5) **Grant/Reject** a request

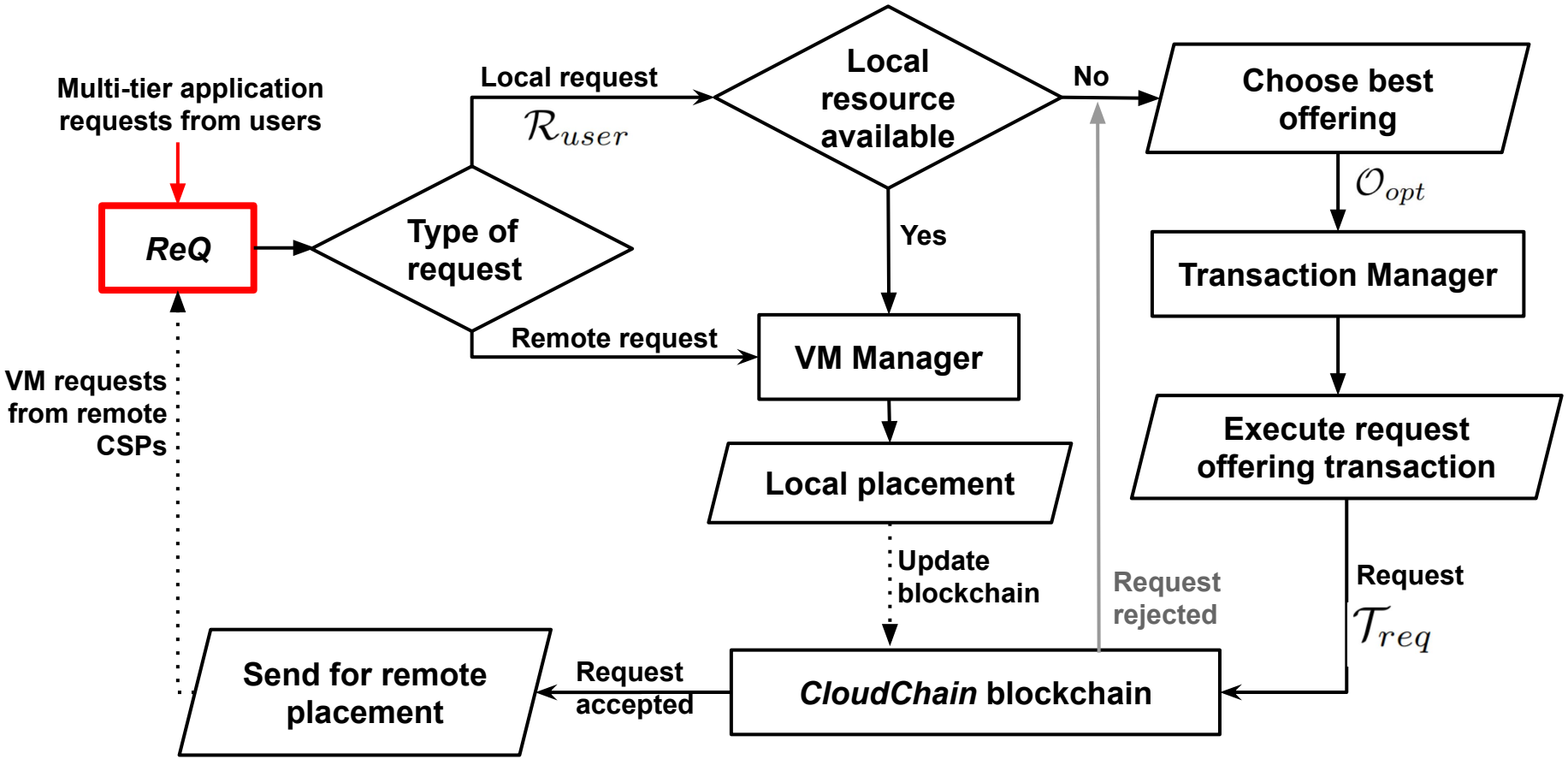
System Flow


Multi-tier Application requests from users
ReQ
Type of Request
Remote Request
Local Request $\mathcal{R}_{user}$
Local resource available
Yes
No
VM Manager
Local Placement
Update Offerings
Request rejected
Choose best Offering
$\mathcal{O}_{opt}$
CloudChain Placement
VM request from remote CSP
CloudChian Blockchain
Transaction Manager
Request $\mathcal{T}_{req}$
Broadcast the chosen request for placement
Request accepted
Send for remote placement
Remote Placement

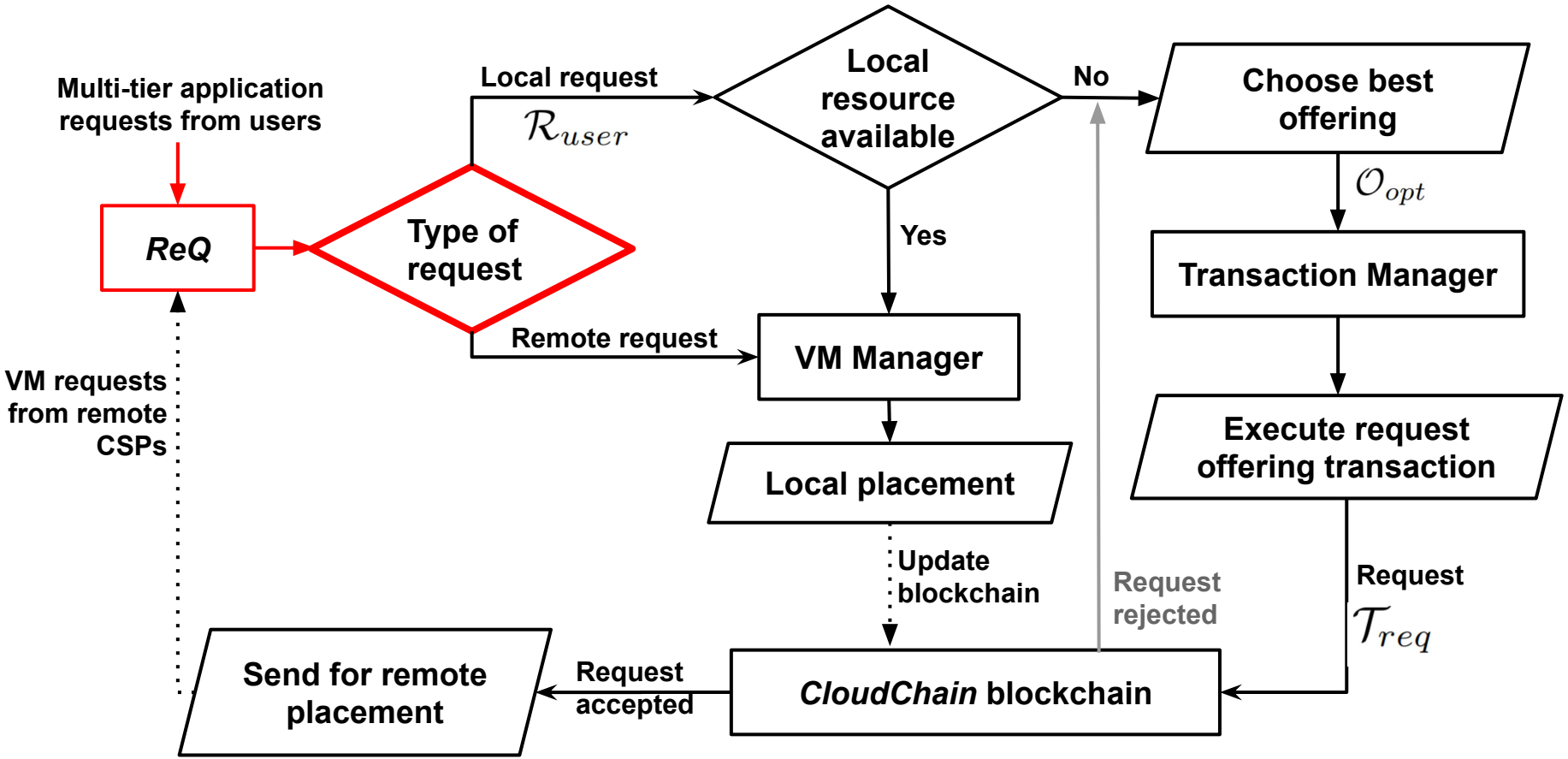
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement
VM requests from remote CSPs

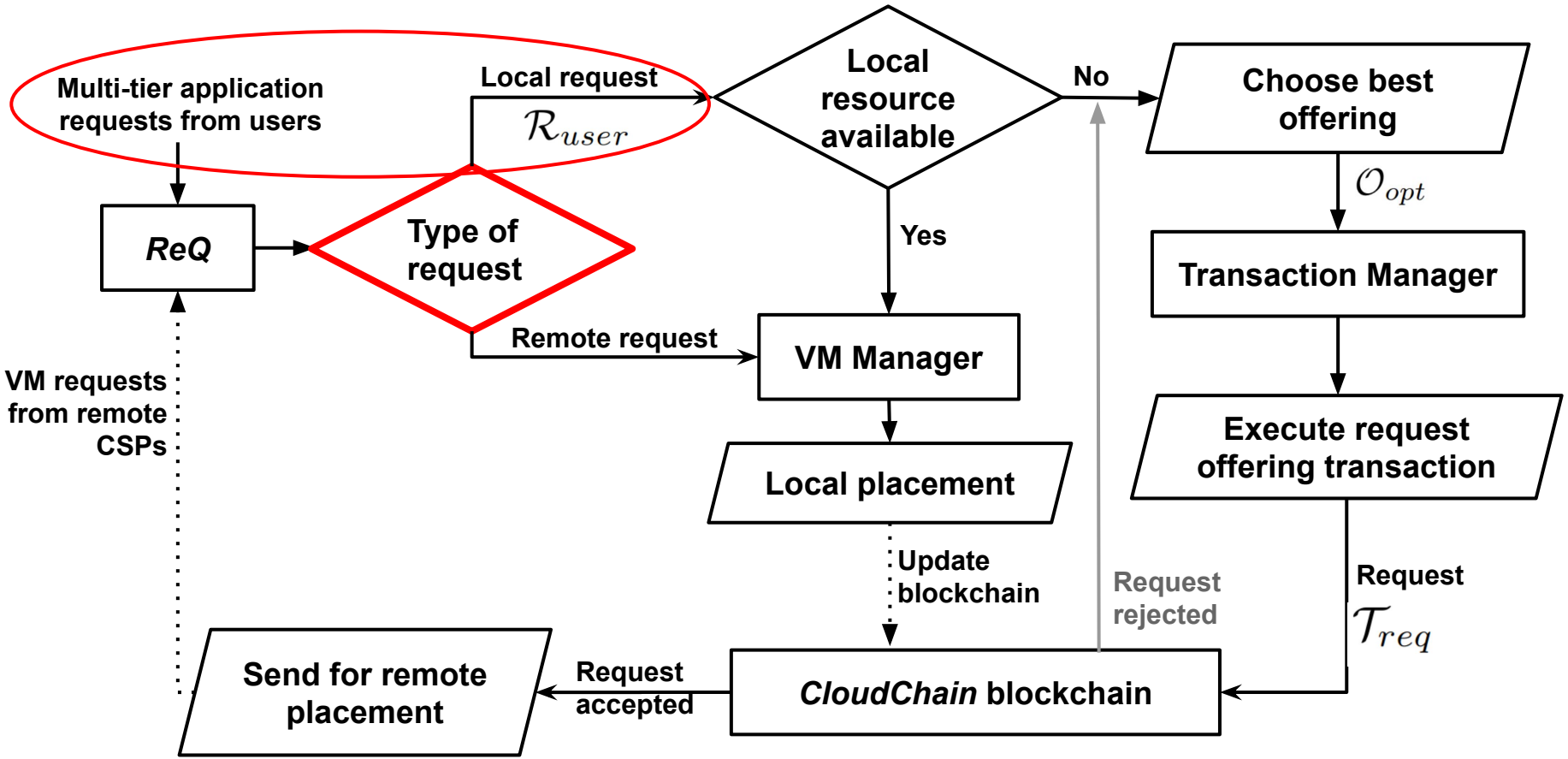
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement
VM requests from remote CSPs

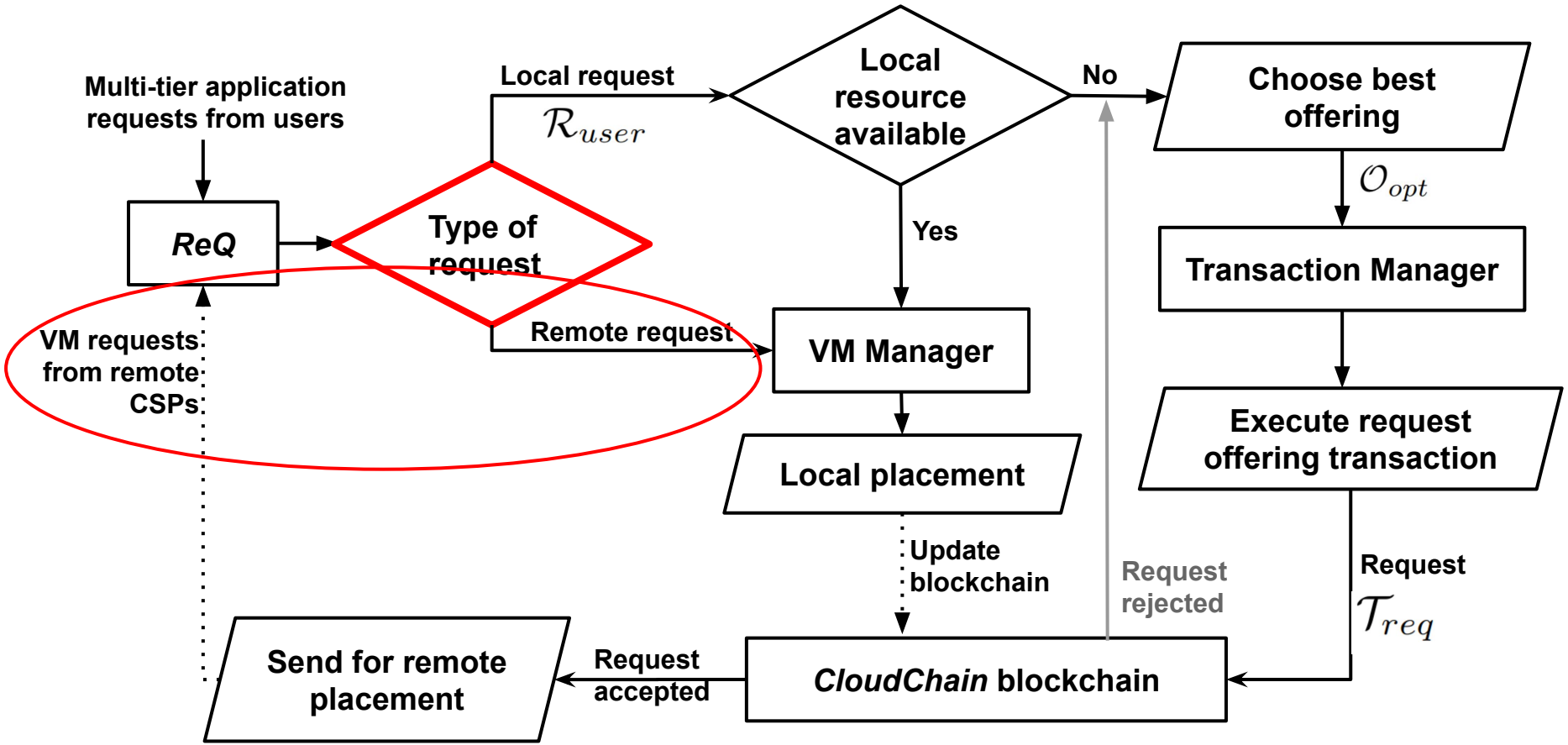
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

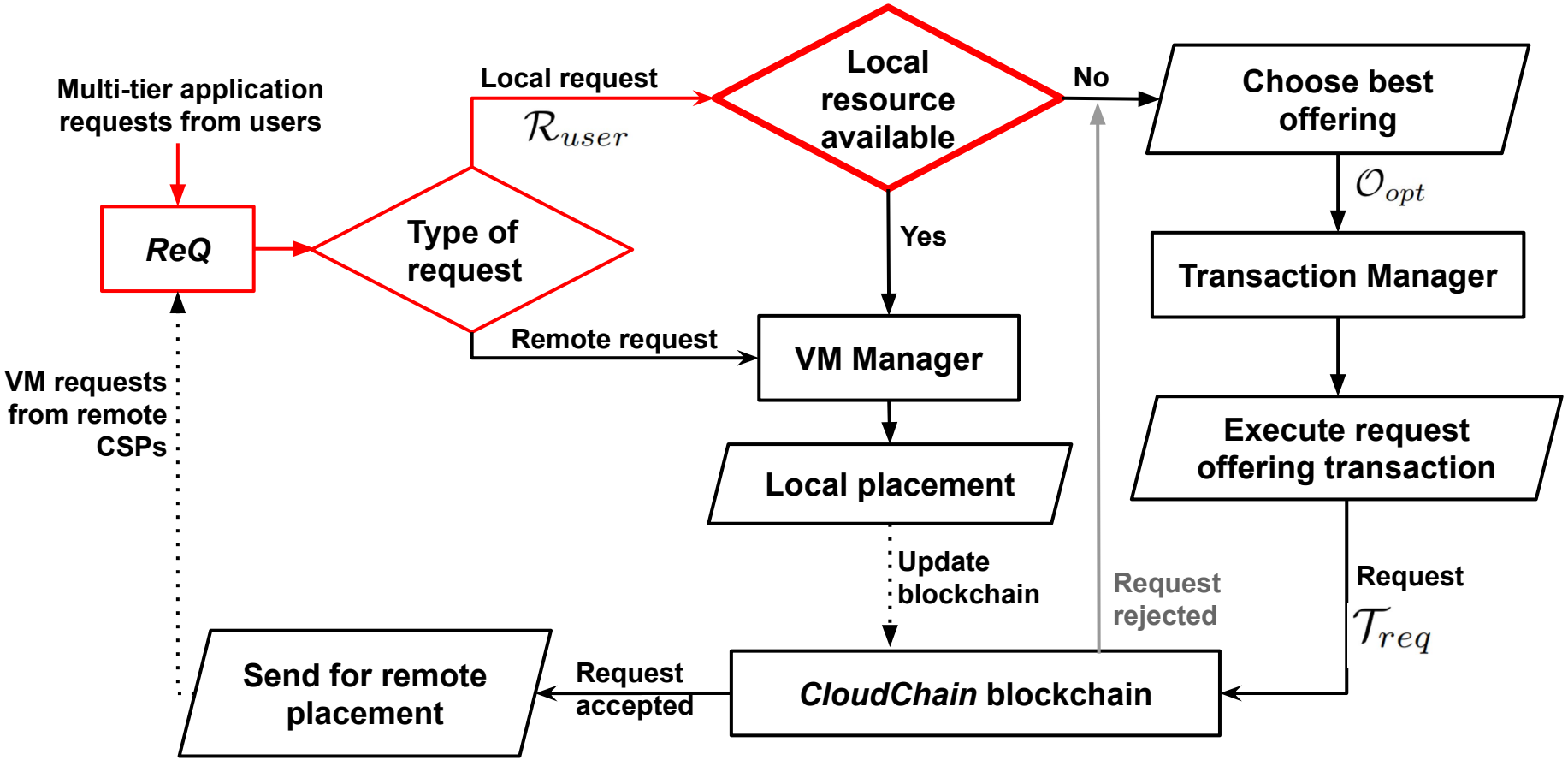
System Flow


Multi-tier application requests from users
Local request
$\mathcal{R}_{user}$
Local resource available
No
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
ReQ
Type of request
Yes
Remote request
VM Manager
Local placement
Update blockchain
Request rejected
VM requests from remote CSPs
Send for remote placement
Request accepted
CloudChain blockchain

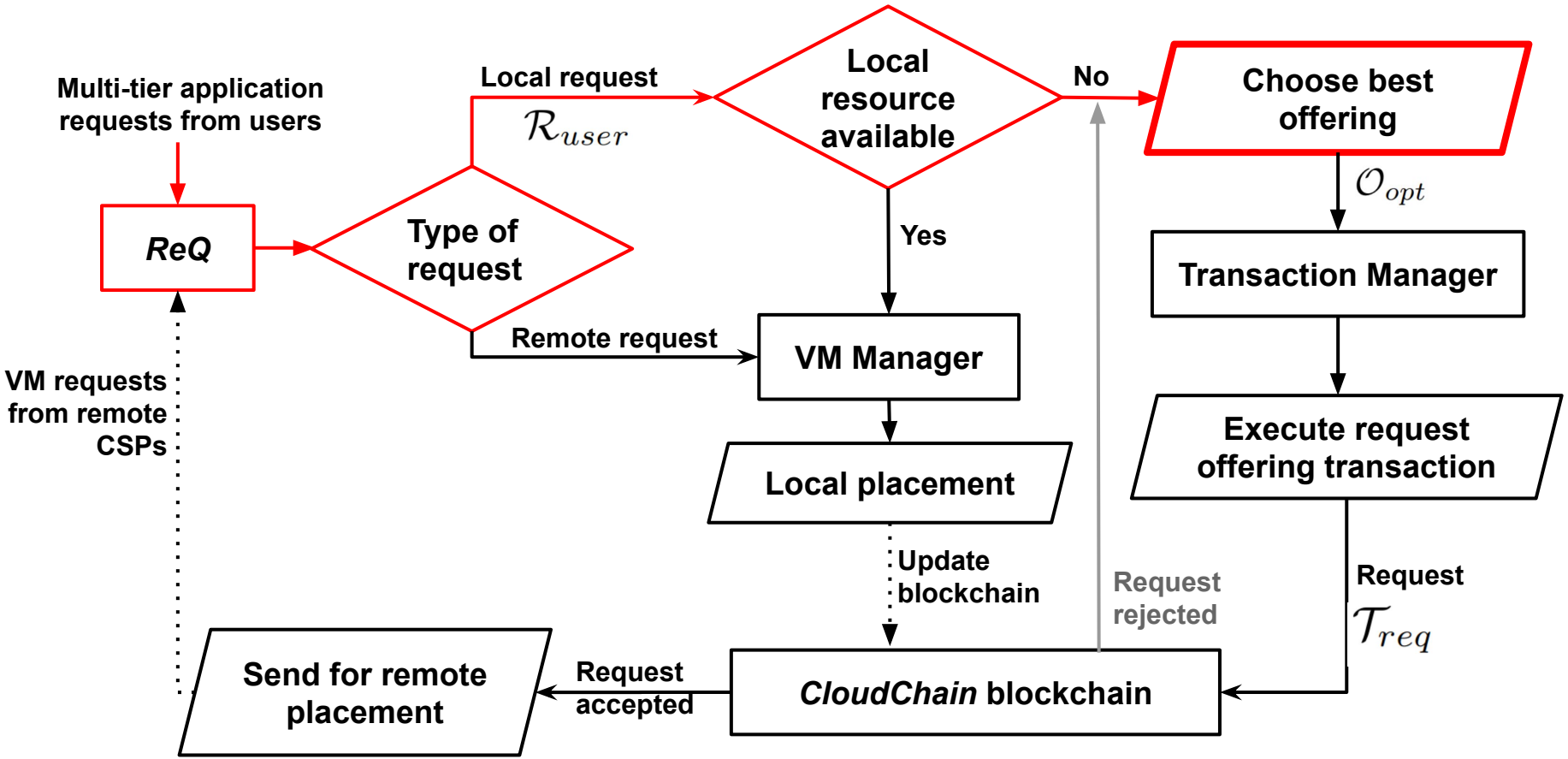
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

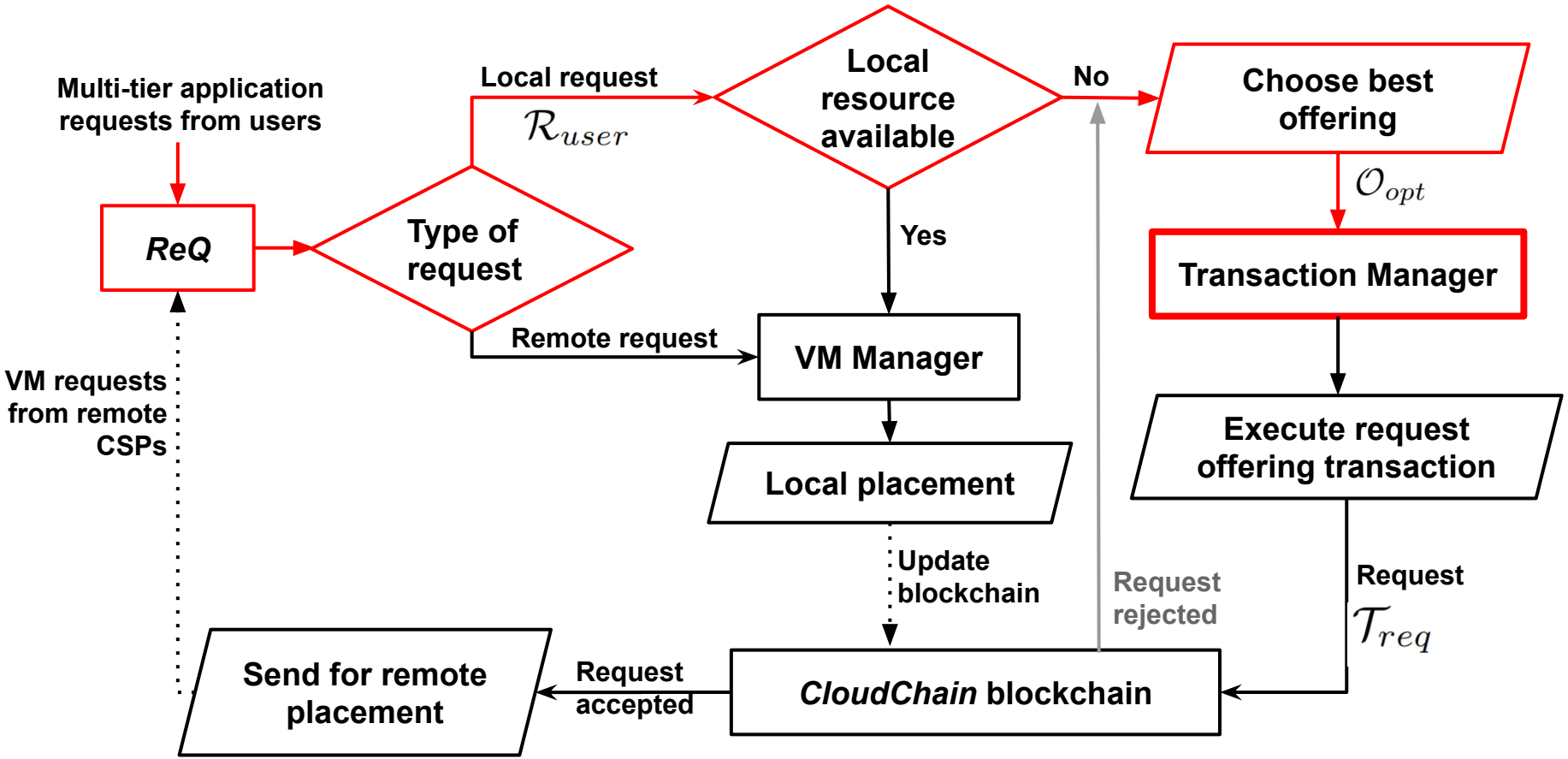
System Flow


Multi-tier application requests from users
ReQ
Local request
$\mathcal{R}_{user}$
Type of request
Remote request
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement
VM requests from remote CSPs

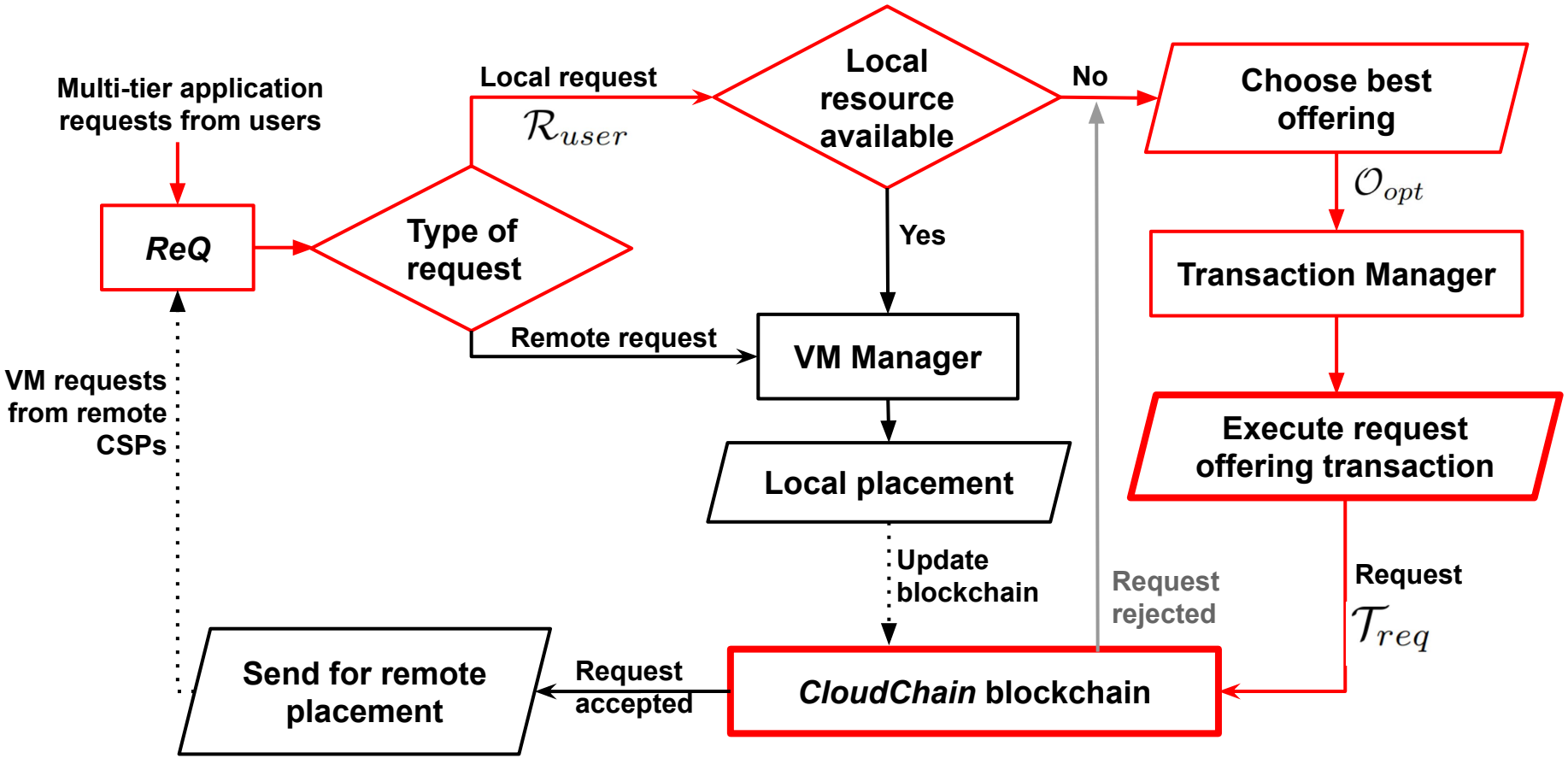
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

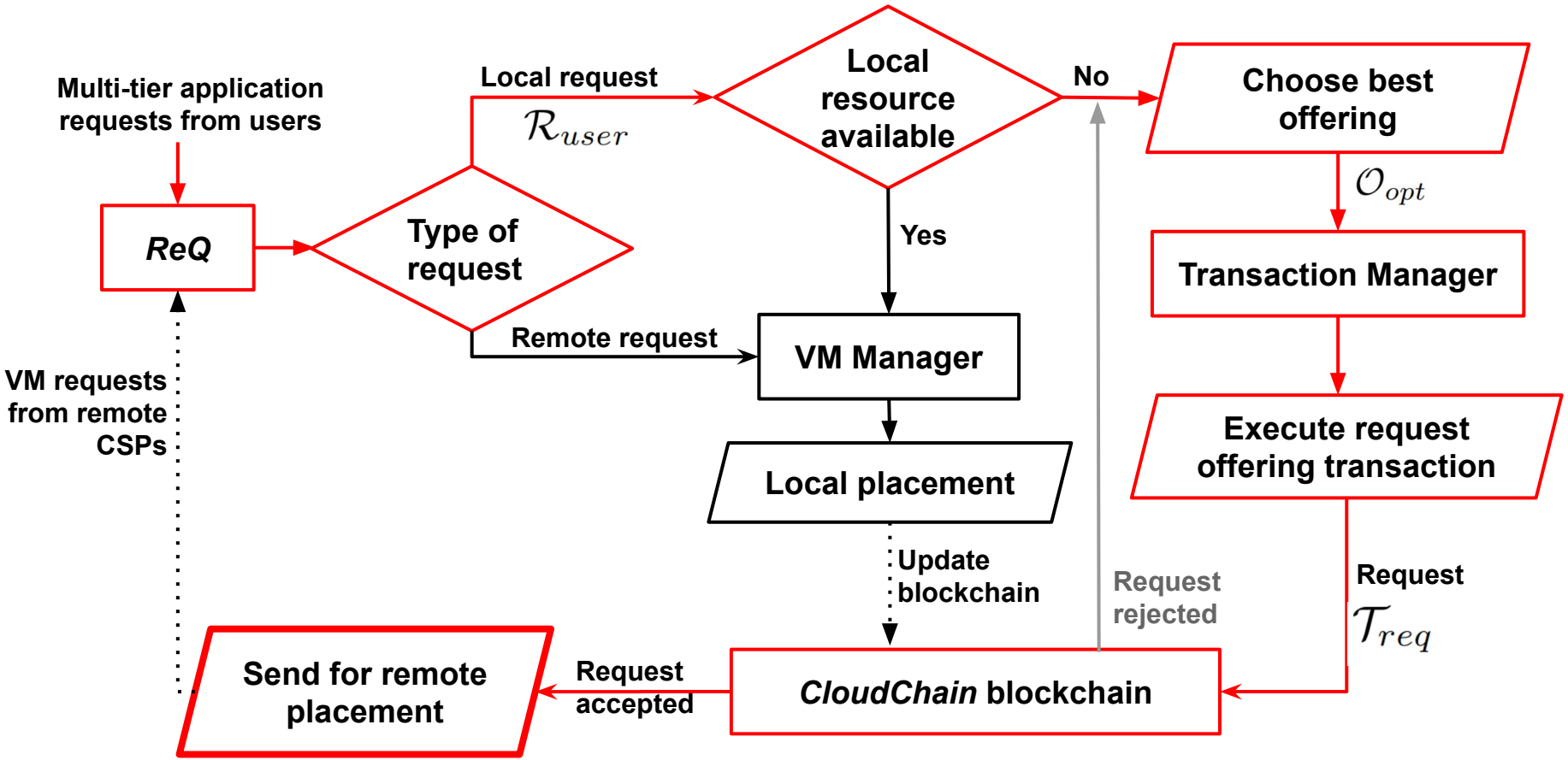
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

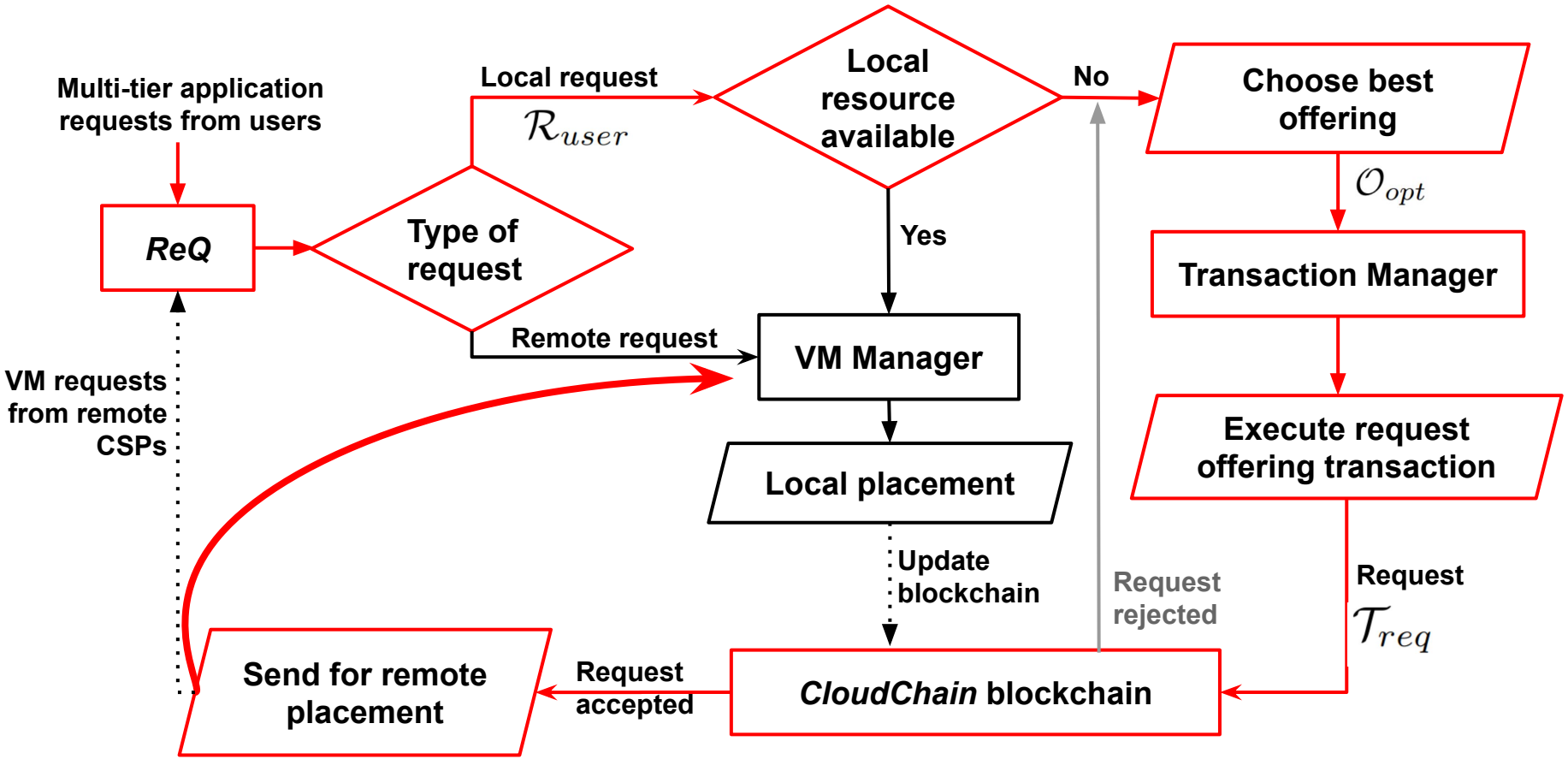
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

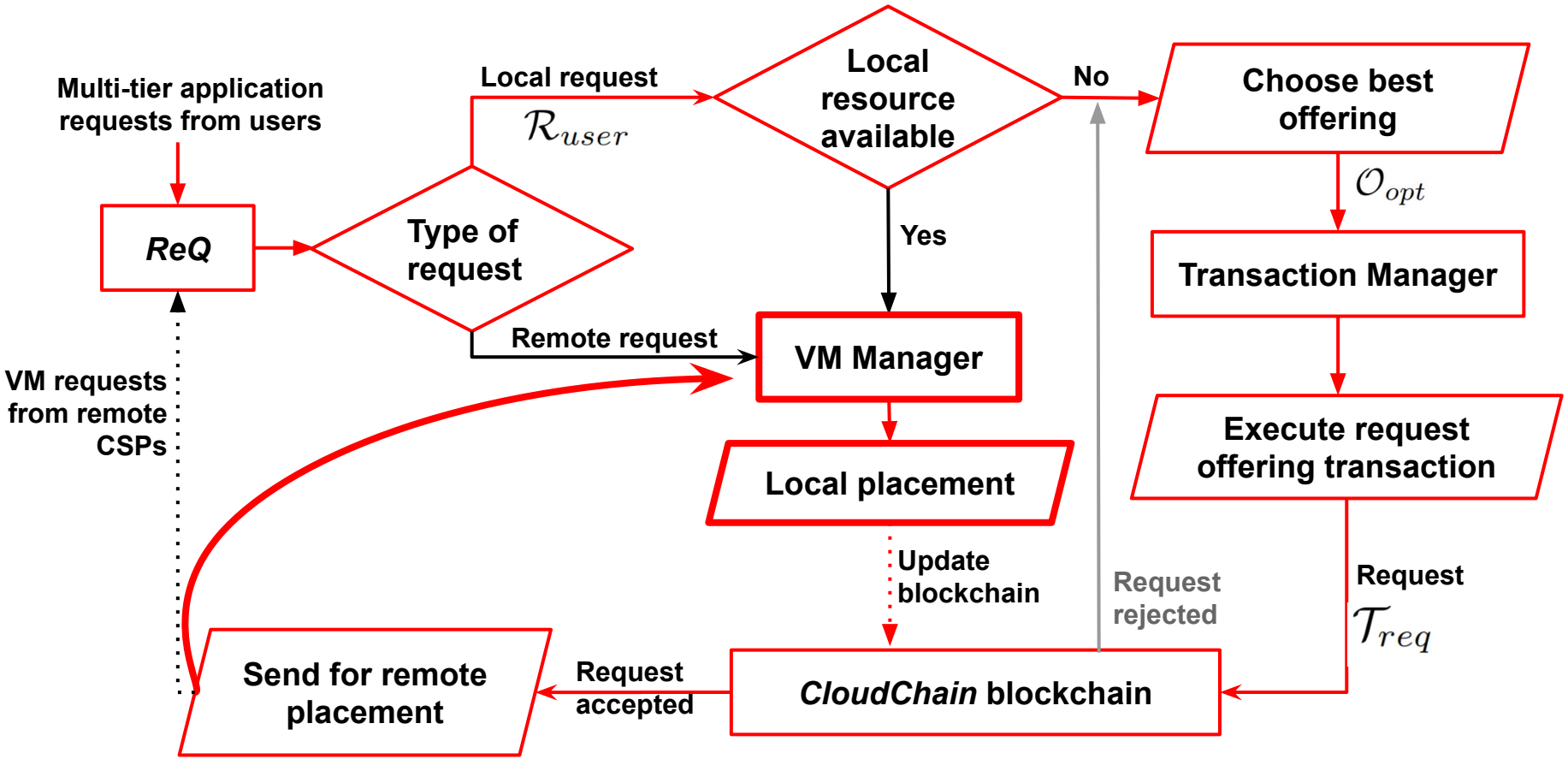
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement
VM requests from remote CSPs

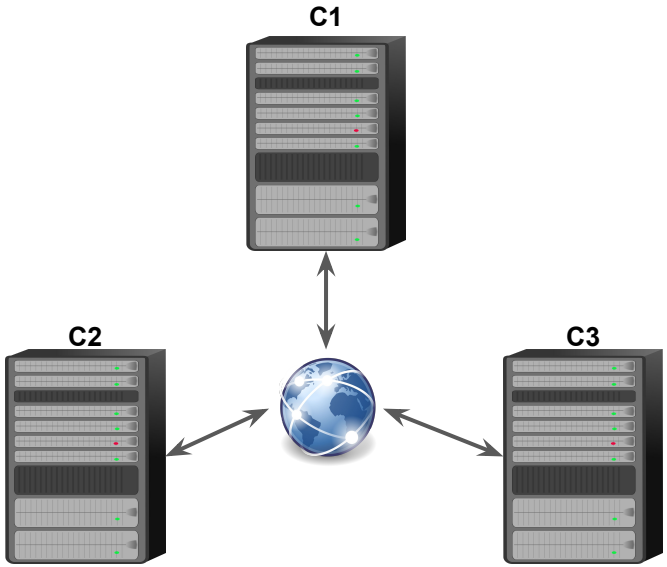
System Flow


Multi-tier application requests from users
ReQ
Type of request
Local request
$\mathcal{R}_{user}$
Remote request
VM requests from remote CSPs
Local resource available
No
Yes
Choose best offering
$\mathcal{O}_{opt}$
Transaction Manager
Execute request offering transaction
Request
$\mathcal{T}_{req}$
VM Manager
Local placement
Update blockchain
Request rejected
CloudChain blockchain
Request accepted
Send for remote placement

Implementation Details

Testbed Setup:

- 3 Hosts, each acting as a cloud connected over the network.



Testbed Parameters.

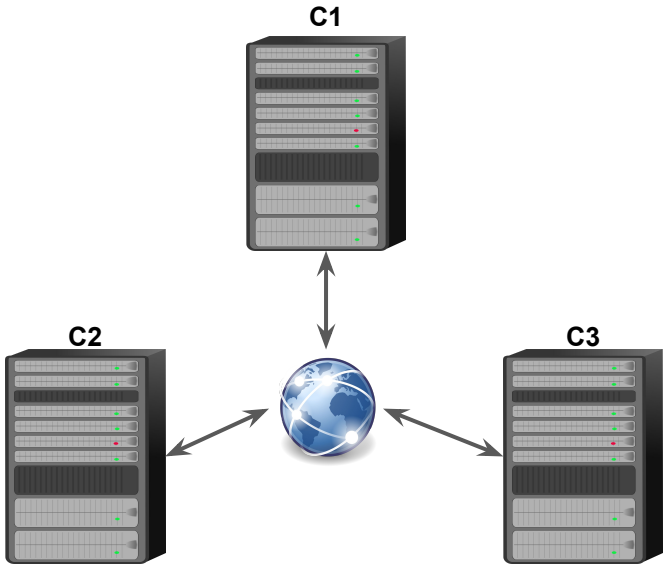
Category	Value
Number of CSP	3
Number of DC per CSP	3
$\mathcal{C}_1$ Config	2.7 GHz, Intel Xeon(R) 48 core, 256 GB Memory
$\mathcal{C}_2$ Config	3.2 GHz, Intel Core i5 4 core, 20 GB Memory
$\mathcal{C}_3$ Config	2.7 GHz, Intel Core i3 4 core, 8 GB Memory
Containerization	Docker 18.06
Language used	Go 1.10, Python 2.7

Implementation Details

Testbed Setup:

- 3 Hosts, each acting as a cloud connected over the network.

- Hyperledger fabric for blockchain (v1.3.0)

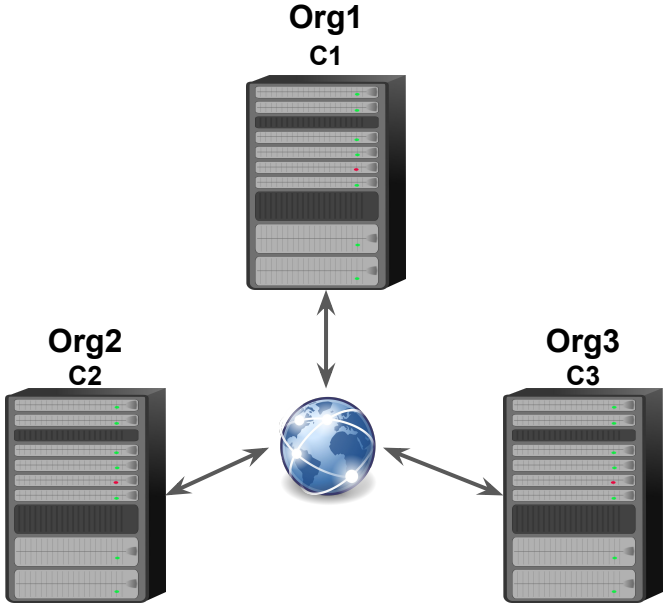


HYPERLEDGER FABRIC

Implementation Details

Testbed Setup:

- 3 Hosts, each acting as a cloud connected over the network.
- Hyperledger fabric for blockchain (v1.3.0).
- Each cloud belongs to a separate **organization**, and runs a **peer.**

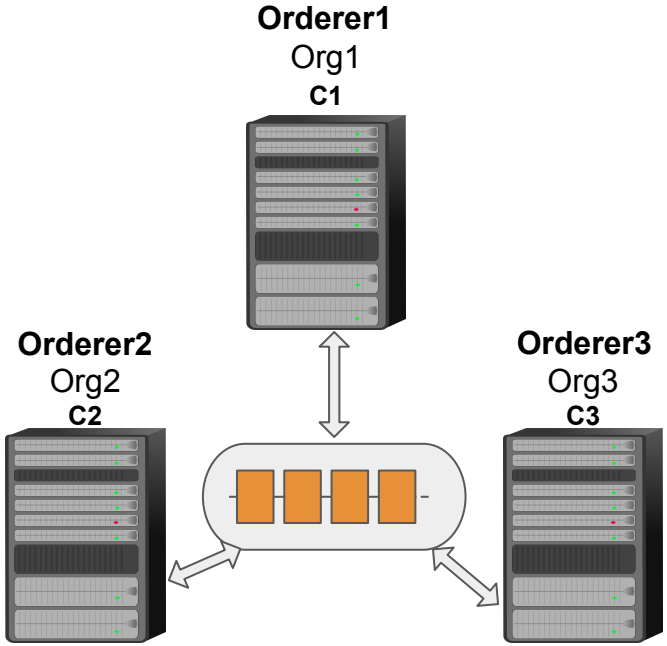


HYPERLEDGER FABRIC

Implementation Details

Testbed Setup:

- Each cloud runs its own **orderer**.

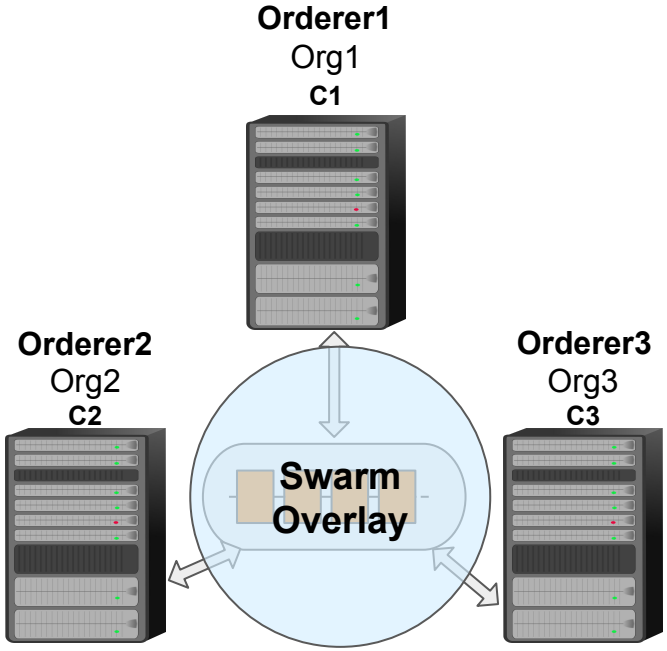


HYPERLEDGER FABRIC

Implementation Details

Testbed Setup:

- Each cloud runs its own orderer
- Create a **docker swarm**.
- Create **overlay network**.

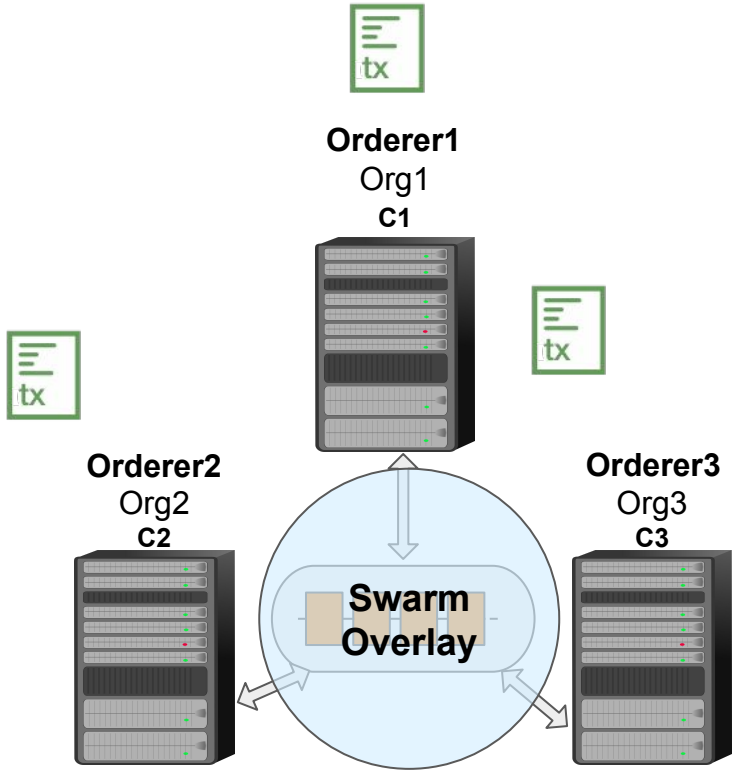


HYPERLEDGER
FABRIC

Implementation Details

Testbed Setup:

- Each cloud runs its own orderer.
- Create a **docker swarm**.
- Create **overlay network**.
- **Chaincodes** for *CloudChain* logic.

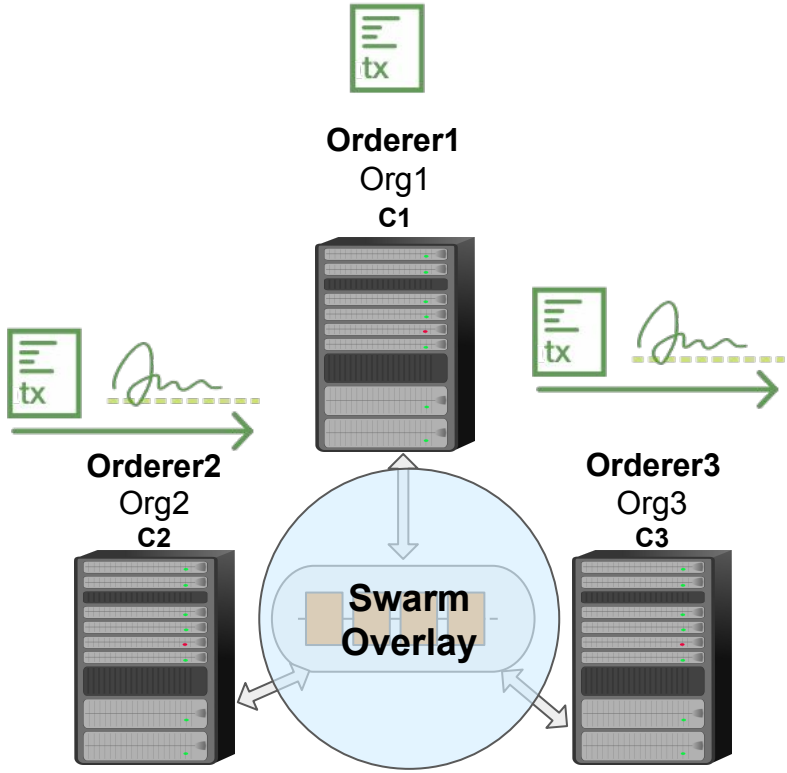


HYPERLEDGER
FABRIC

Implementation Details

Testbed Setup:

- Each cloud runs its own orderer.
- The orderers use **BFT** protocol.
- **Chaincodes** for Cloud Exchange logic.
- **Endorsement policy** requiring the endorsement of the concerned demanding SP, supplying SP and the majority of other endorsing peers.

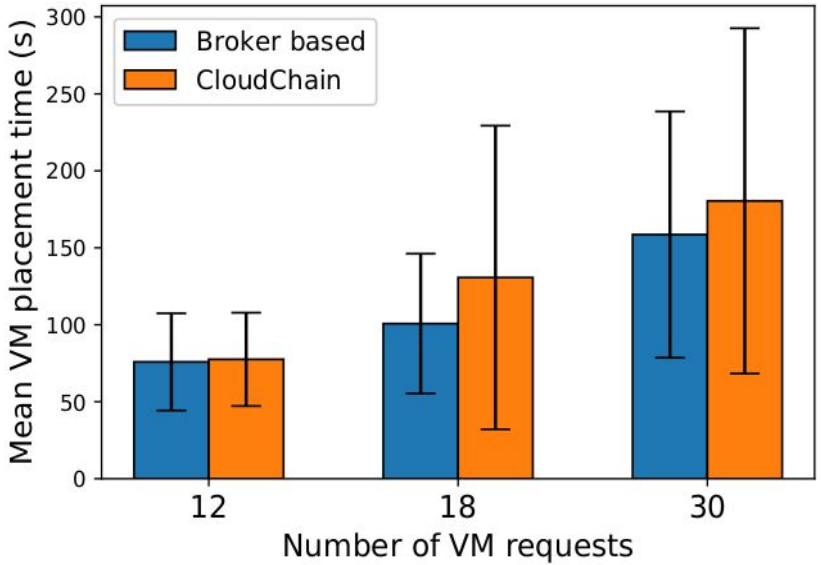


HYPERLEDGER
FABRIC

Results

- Mean VM placement time in broker based federation and *CloudChain*
- Three scenarios
- Each CSP receives 4, 6, and 10 VM requests in first, second and third scenarios respectively.
- In case of broker based federation all the requests arrive at the broker first.

Mean VM placement time

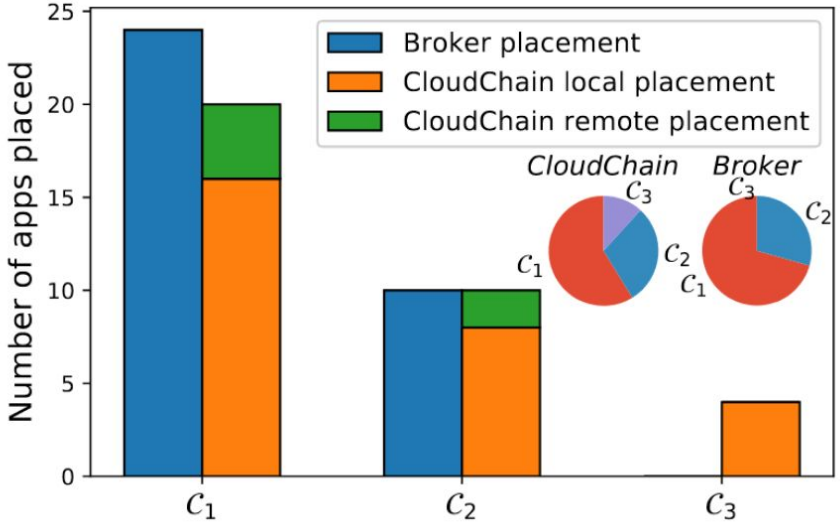


Very little compromise in performance

Results

- 34 multi-tier application requests
- *C1*, *C2* and *C3* receiving 16, 8 and 10 requests respectively.
- In case of broker based federation, all requests arrive at the broker

Distribution of user requests across different CSPs



C3 is starved in case of broker based system

***CloudChain* shows fair distribution**

Conclusion and Future work

***CloudChain* over Federation brokers:**

- Decentralized
- Transparent
- Autonomy
- Immutable
- Fairness

Conclusion and Future work

Future work

- Support for live VM migration
- FLA monitoring

Acknowledgements

Special thanks to:

Microsoft Research Travel Grant.

CNeRG for Travel Grant and constant support. (cnerg.iitkgp.ac.in)

Thank You

Bishakh Chandra Ghosh (ghoshbishakh@iitkgp.ac.in), Sourav Kanti Addya, Anurag Satpathy, Soumya K Ghosh and Sandip Chakraborty