



# DCAS & Gotary

P. Bächli, D. Thoma & L. Röllin

**DCAS**

—



# DCAS Principles

Distributed Communication & Authorization System => IM

Written in Kotlin

Based on TomP2P

All operations on DHT tables

Everything is a group chat





# DCAS Operations

Authorization

Contact List

Chat messages in DHT

Poll DHT periodically (~1s)

Online state



# Tables

Chat membership (table per nickname)

| user_chats_\$nickname |
|-----------------------|
| ChatId1               |
| ChatId4               |
| ChatId7               |

User online status (table per nickname):  
Last online timestamp, offline if >30  
minutes

| user_online_\$nickname   |
|--------------------------|
| 2018-12-11T08:10:00+0000 |



# Tables

Chat messages (table per chat ID)

| chat_messages_\$chatId      |
|-----------------------------|
| Pair<Timestamp, MessageDto> |
| Pair<Timestamp, MessageDto> |
| Pair<Timestamp, MessageDto> |

MessageDto:

- Sender nickname
- Recipient chat ID
- Message body

# Gotary

---



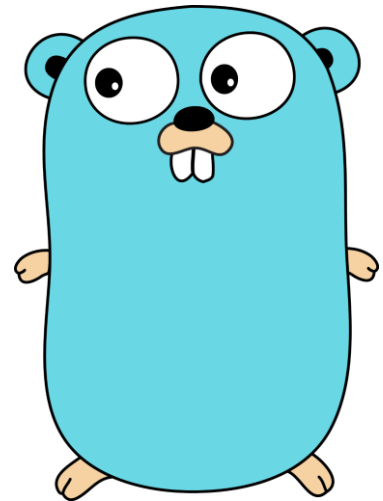
# Gotary

Notary Service

Ethereum / Solidity

Go

CLI







# Maps

Hash => Sender

| Hash | Sender<br>Public Key |
|------|----------------------|
| 0x1  | 0x7E8e1147DeA        |
| 0x3  | 0x1fa96FcC0C8        |
| 0x5  | 0x4699Dd2cb88        |

Hash => Recipient

| Hash | Recipient<br>Public Key |
|------|-------------------------|
| 0x1  | 0x40a9a634a32           |
| 0x3  | 0x18a494cDC36           |
| 0x5  | 0xe6148AC056<br>B       |



# State Map

| Hash | State   |
|------|---------|
| 0x1  | <State> |
| 0x3  | <State> |
| 0x5  | <State> |

```
enum Status {  
    PENDING,  
    ACCEPTED,  
    REJECTED  
}  
  
struct State {  
    Status status;  
    bool existent;  
}
```

# Lessons learned

---



## Lessons learned

Asynchronous testing is hard

TomP2P: nice abstraction

Smart Contracts seem not so difficult

... but difficult to get it right