



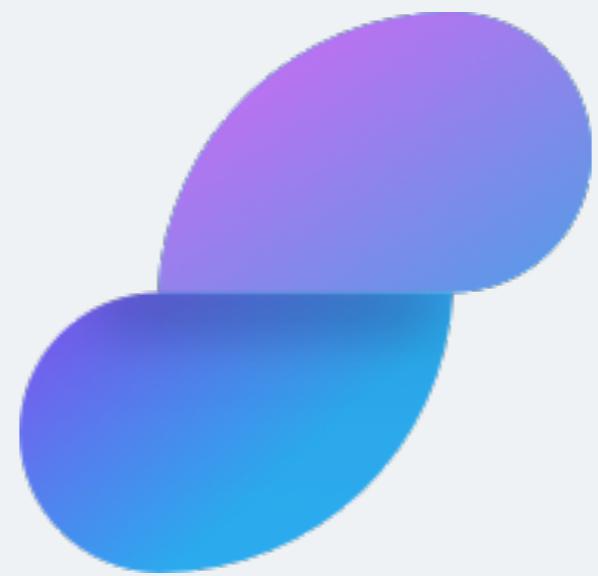
# Status

Integrating Ethereum into our daily lives

(an Ethereum light client for Android & iOS)

<https://status.im>



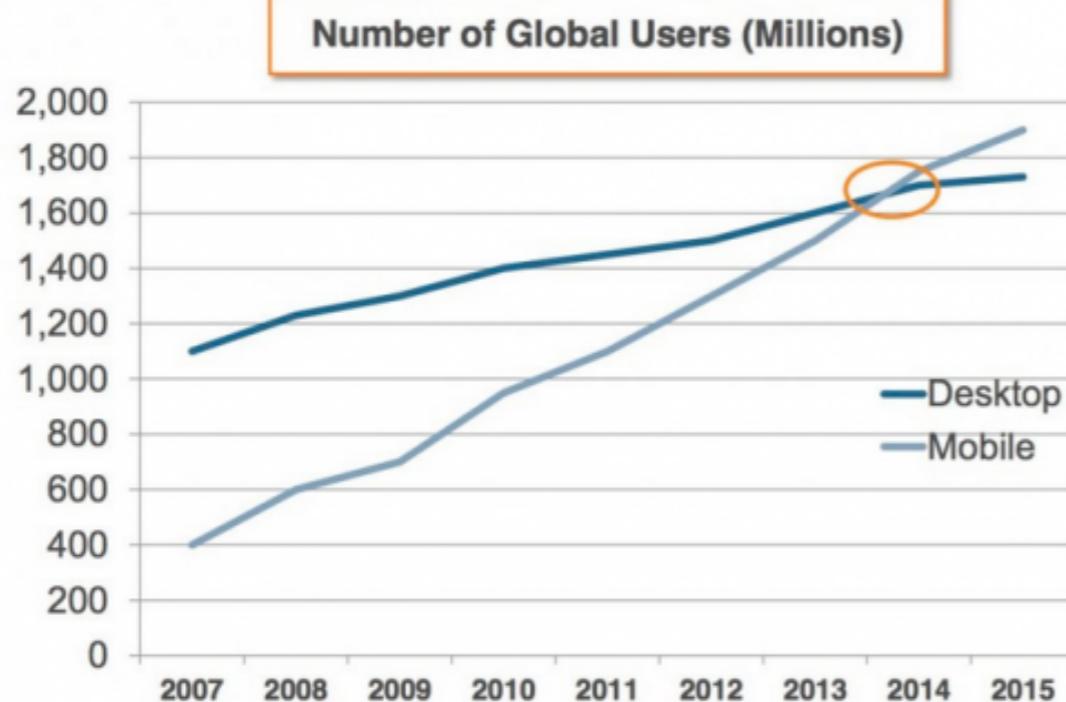


# WHY ETHEREUM?

a foundation for a programmable social system  
operational closures for experimentation  
non-physical free trade zones



# Smartphones are the new PC



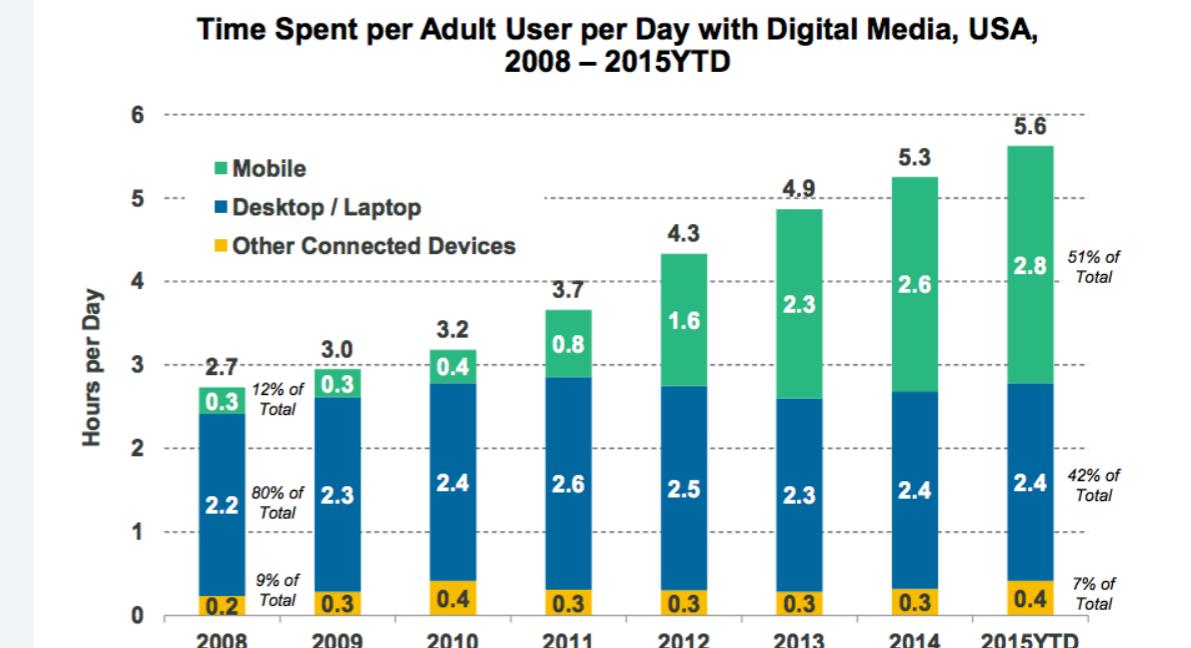
comSCORE

© comScore, Inc. Proprietary and Confidential.

24

Source: Morgan Stanley Research

**Internet Usage (Engagement) Growth Solid  
+11% Y/Y = Mobile @ 3 Hours / Day per User vs. <1 Five Years Ago, USA**



@KPCB

Source: eMarketer 9/14 (2008-2010), eMarketer 4/15 (2011-2015). Note: Other connected devices include OTT and game consoles. Mobile includes smartphone and tablet. Usage includes both home and work. Ages 18+; time spent with each medium includes all time spent with that medium, regardless of multitasking.

14

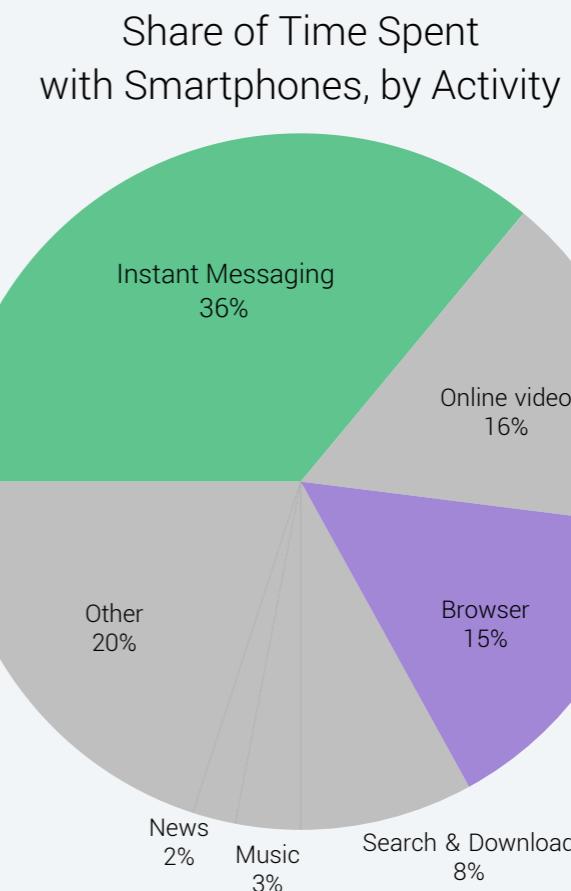
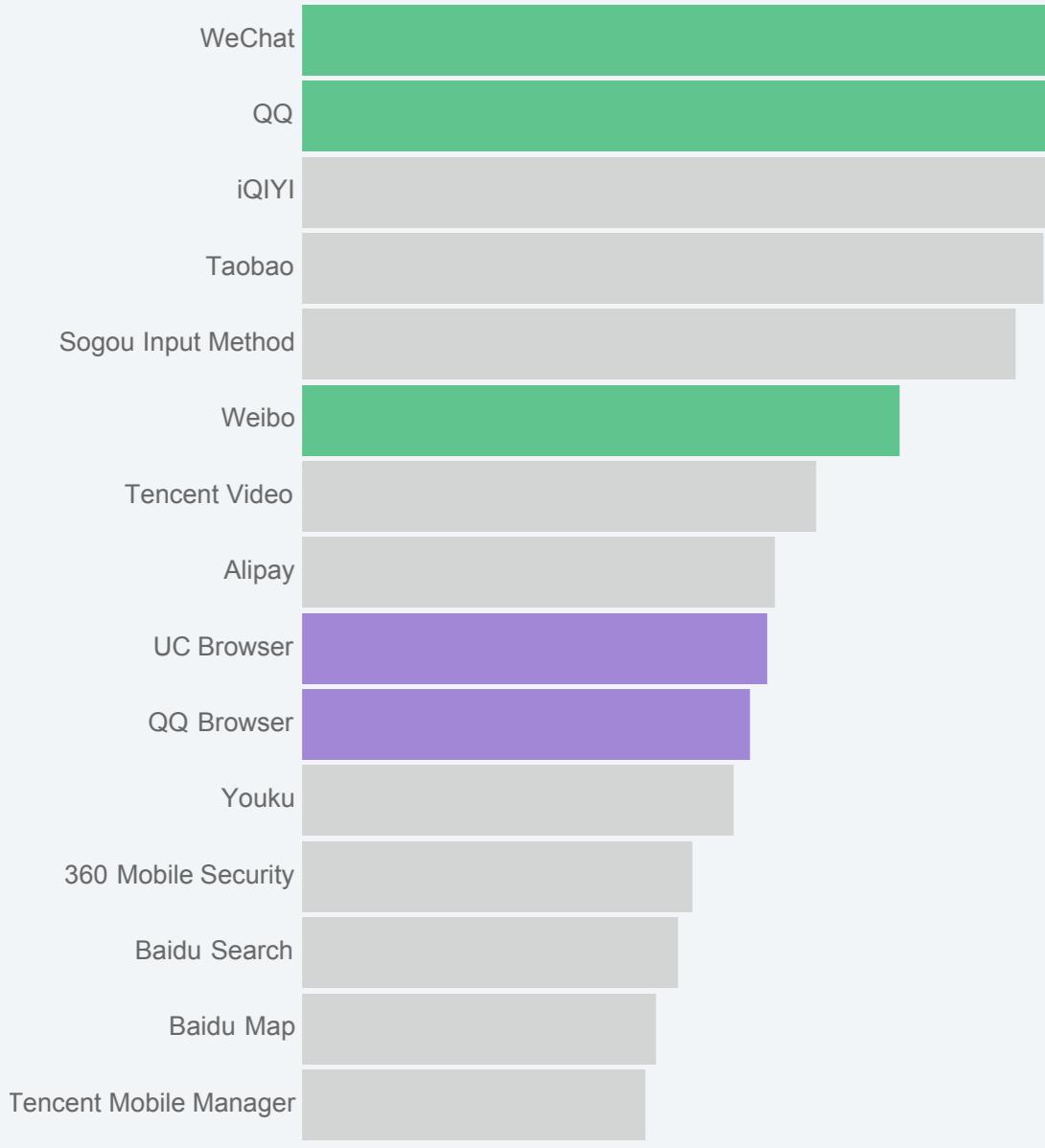
# 2015: "Mist for Mobile"

## 2016: ....



# How are people using Smartphones? 🤔

Top Ranking Apps by Monthly Active Users



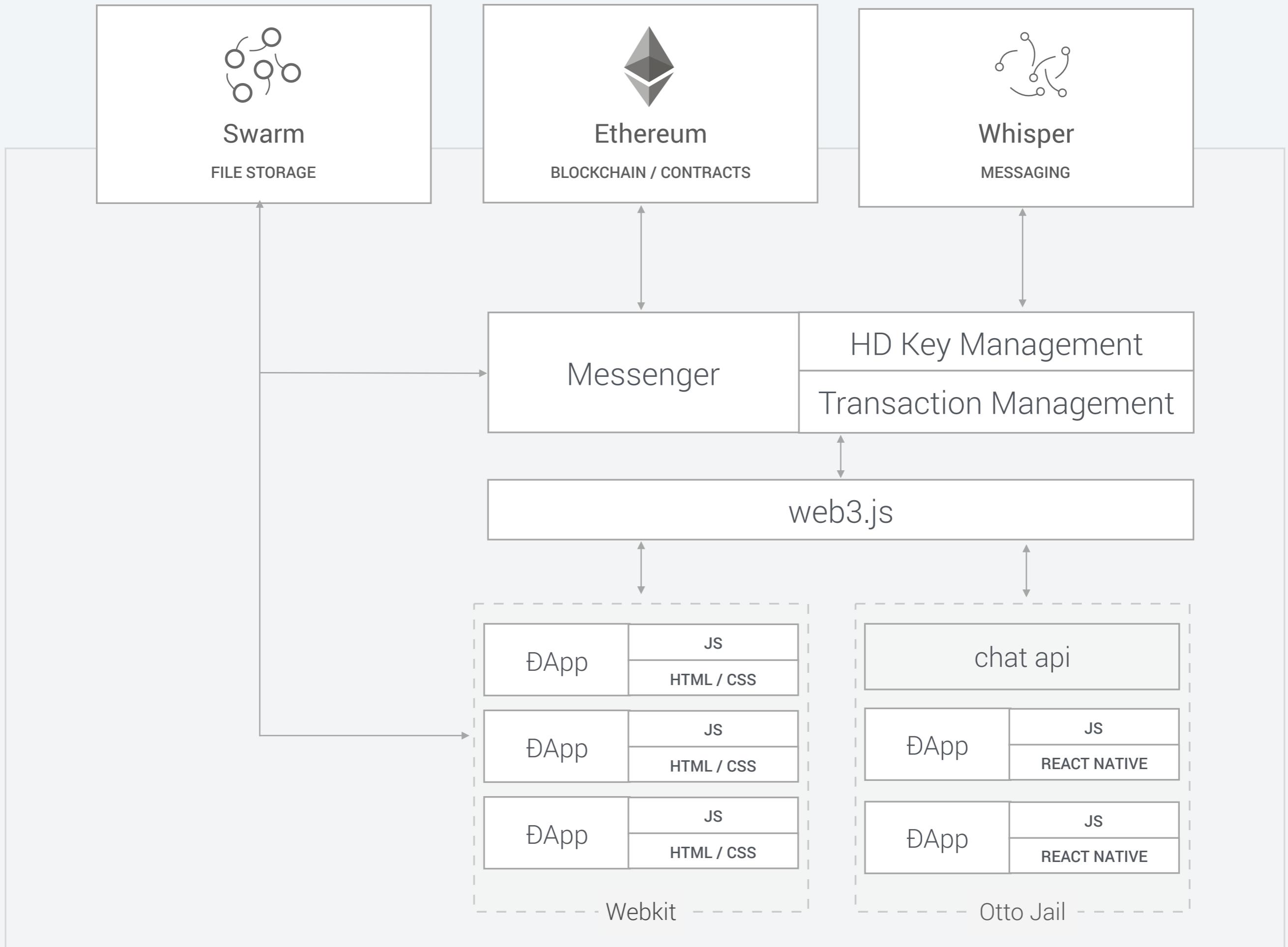
Source:

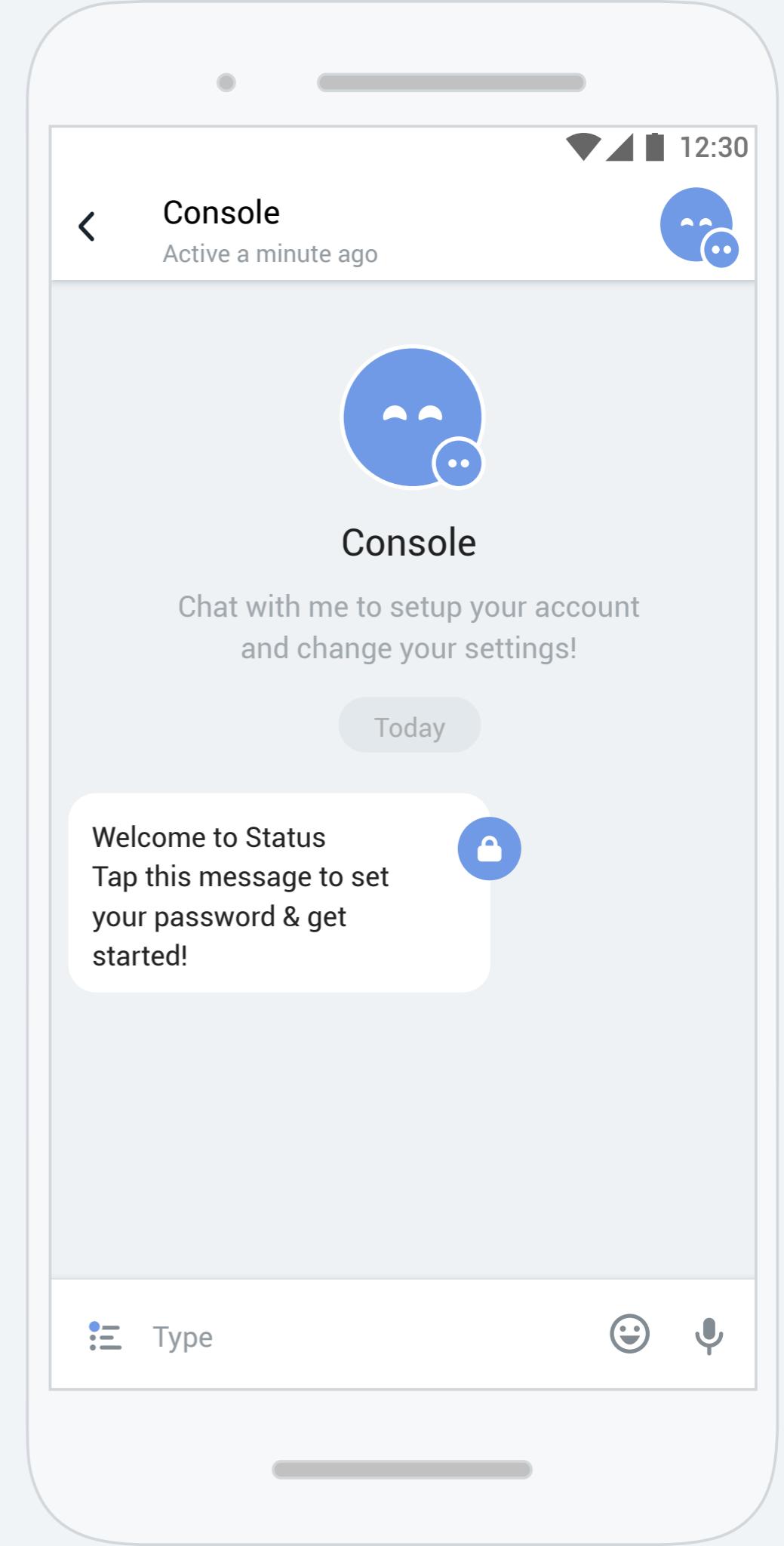
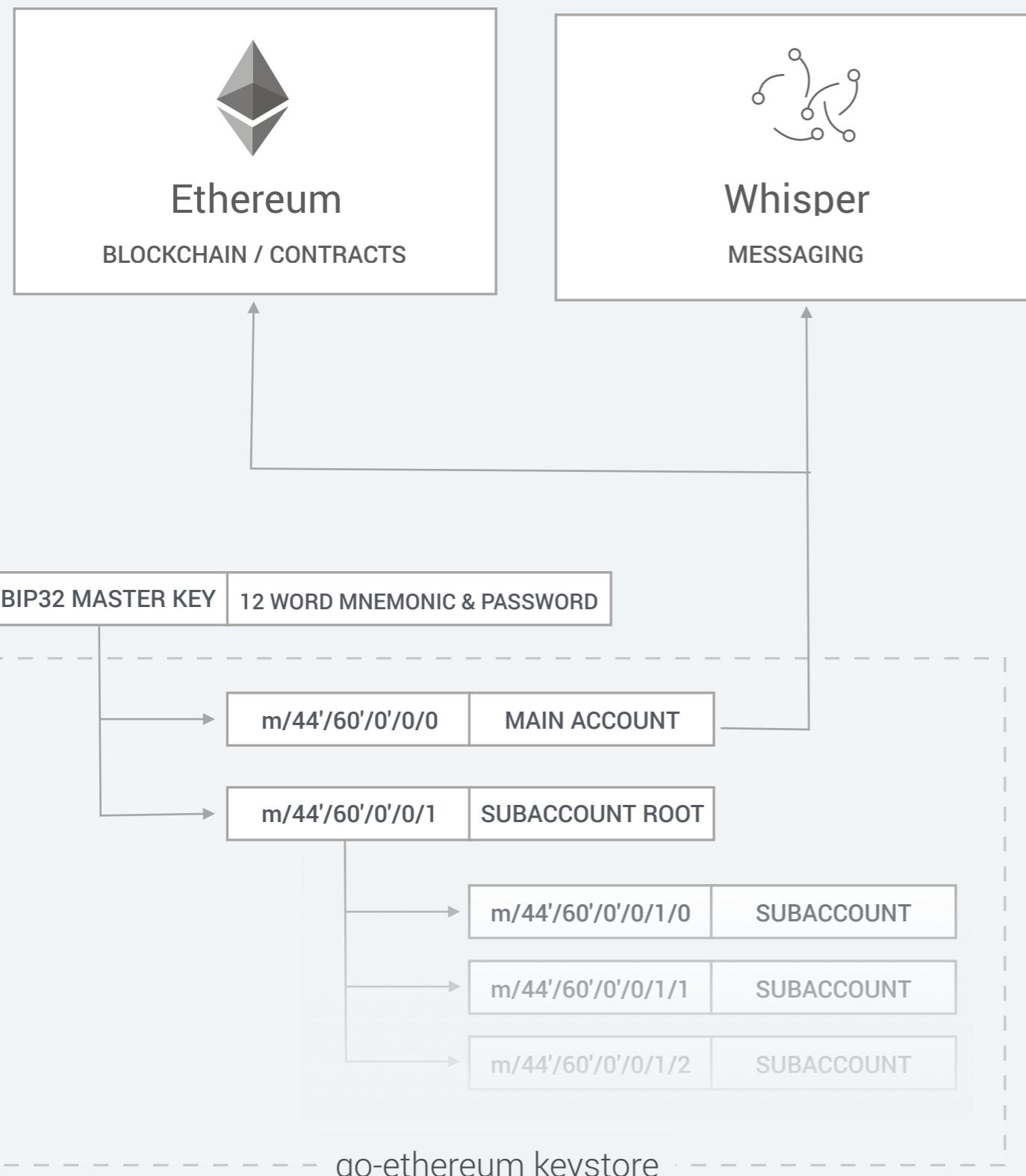
<http://www.vertoanalytics.com/verto-analytics-and-iresearch-partner-to-measure-the-chinese-internet-ecosystem/>  
QuestMobile "China Mobile Internet 2015 Annual Report", Jan 6, 2016

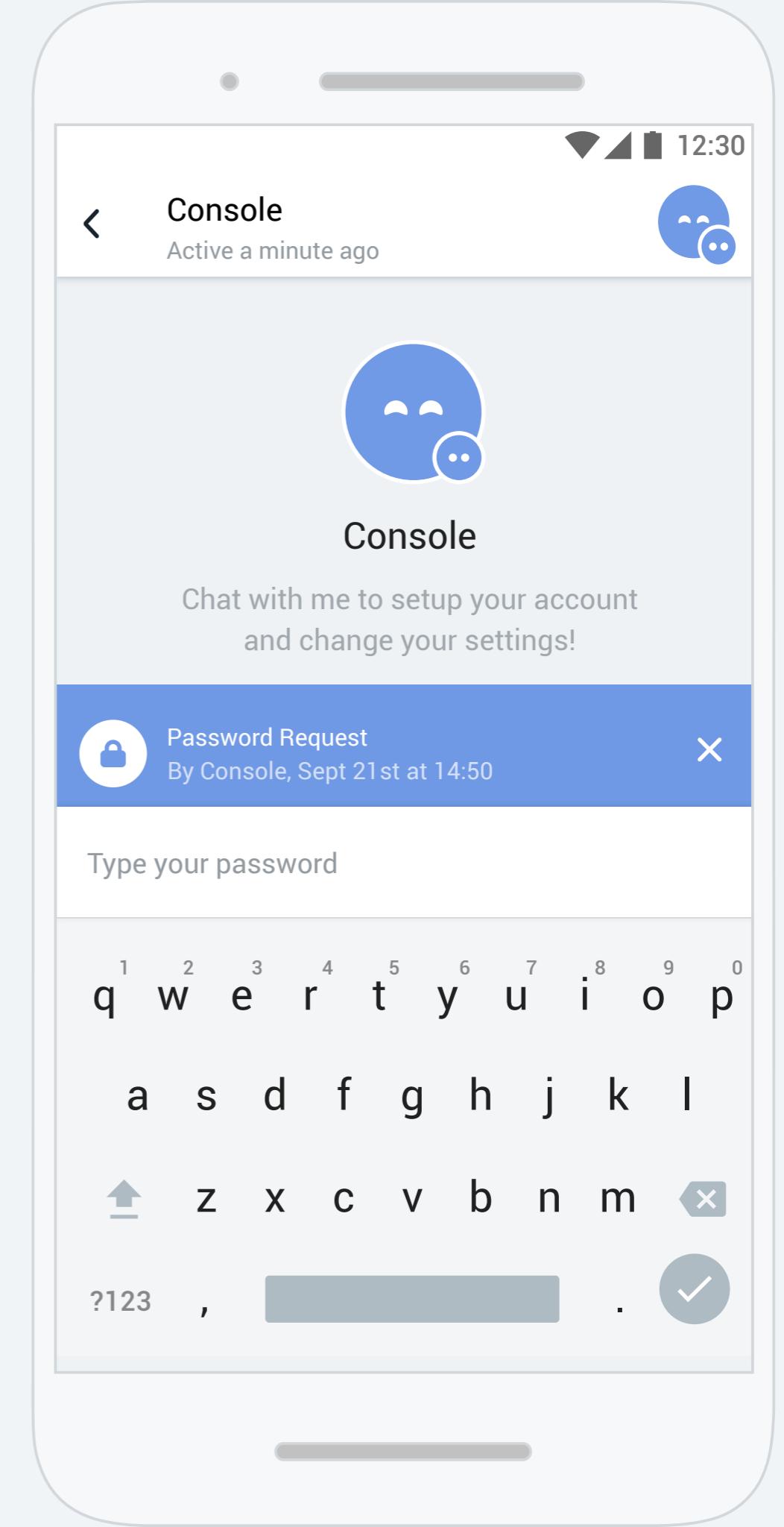
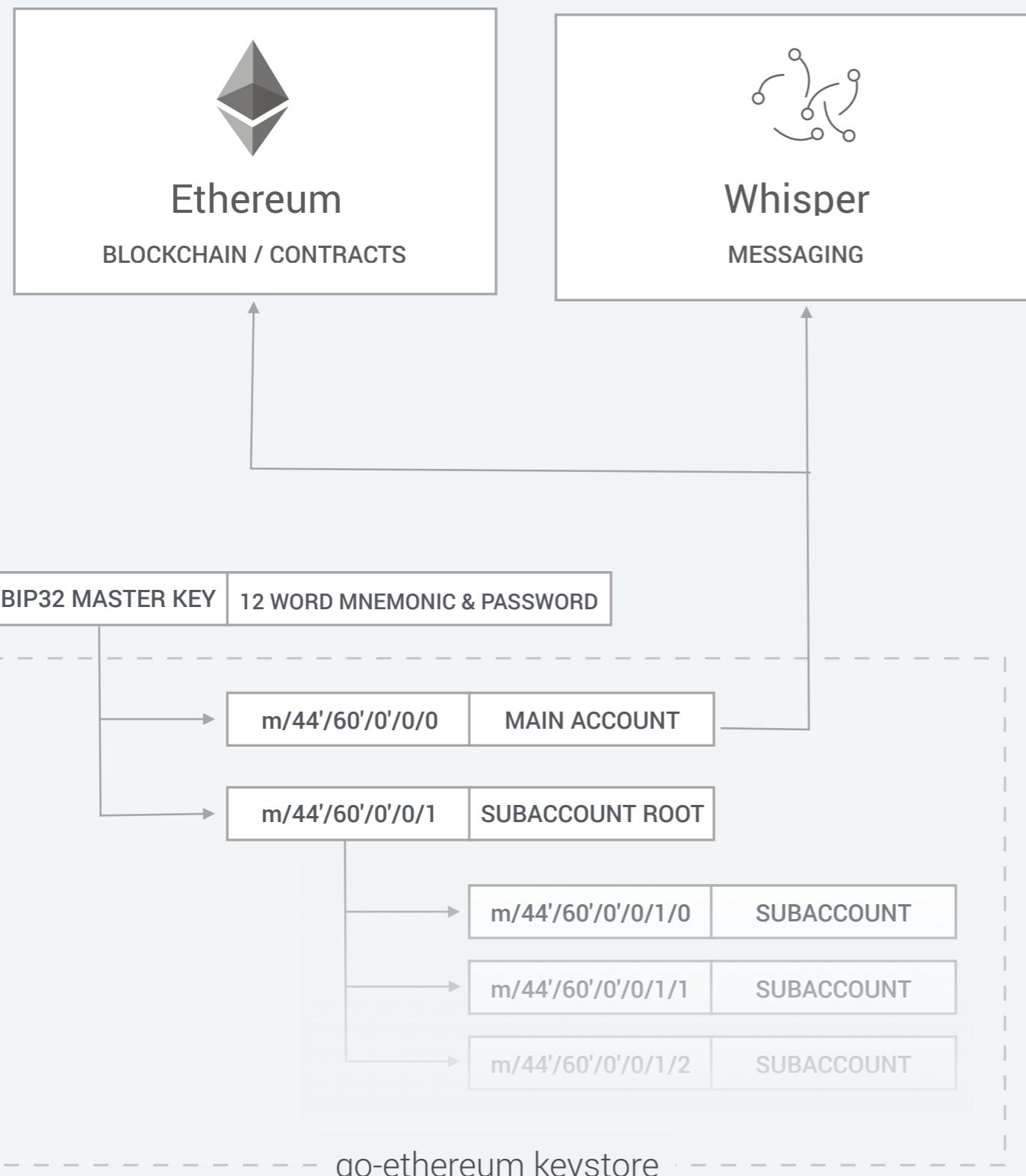


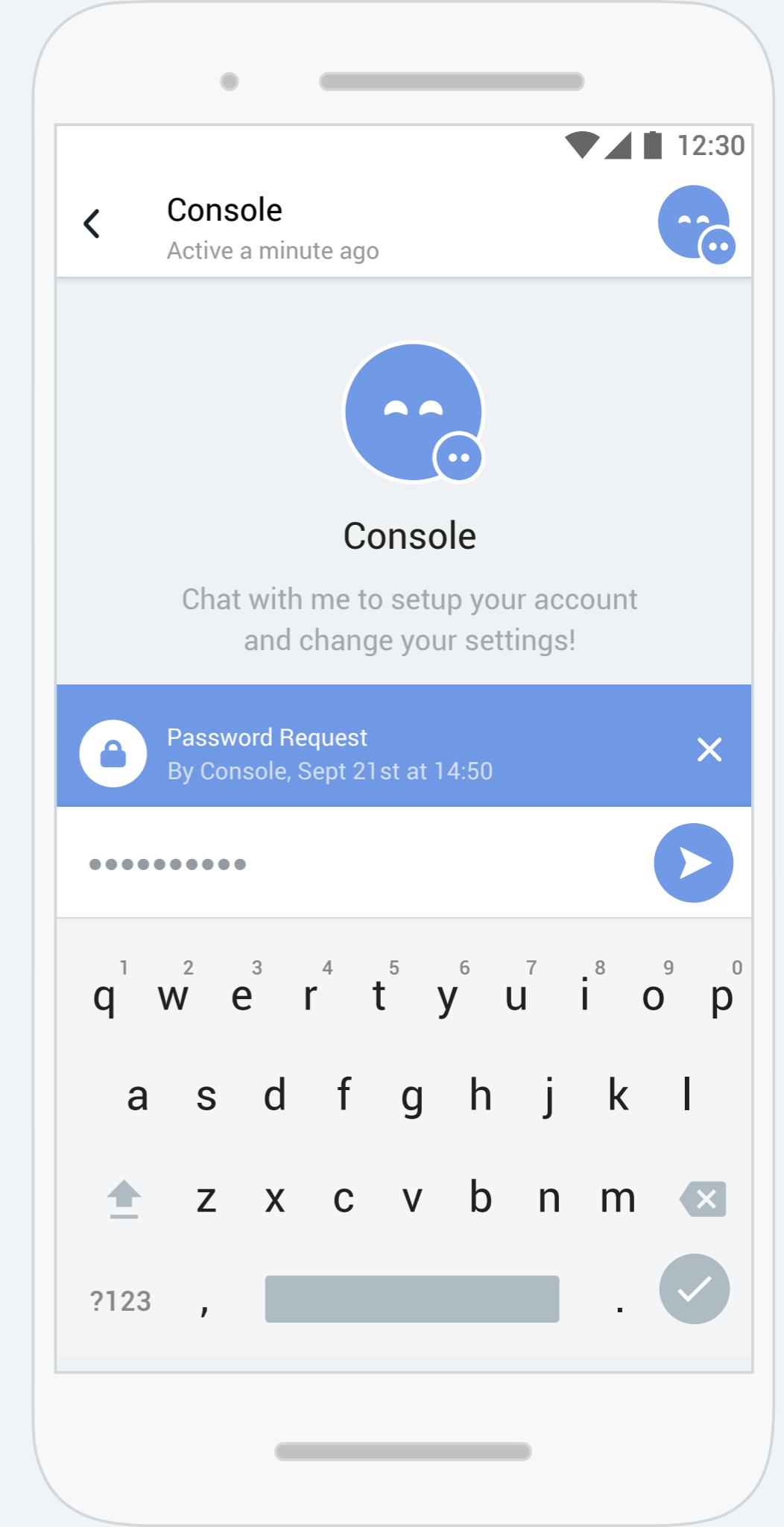
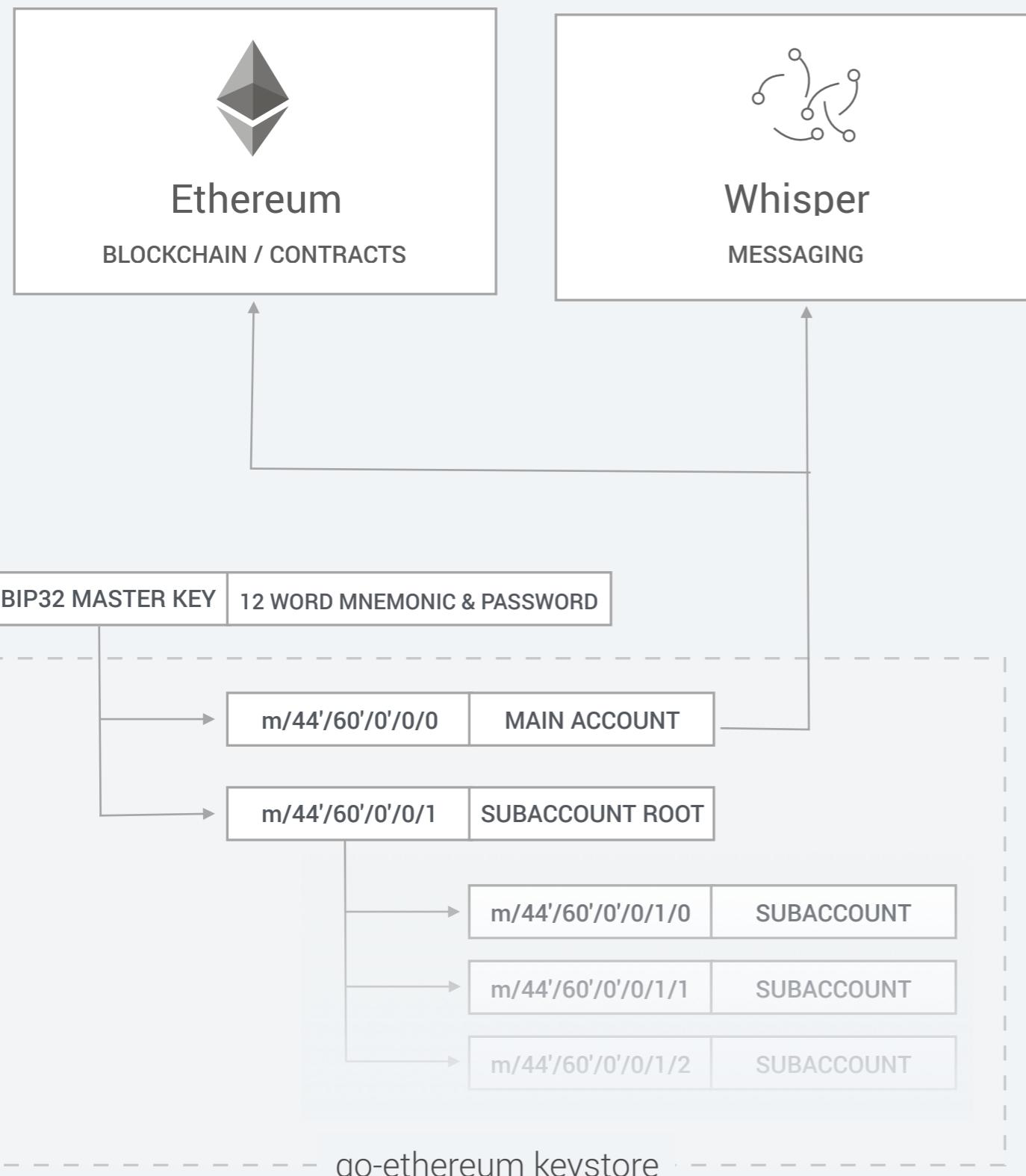
pixelated 🧨 humanise the protocol 🙌

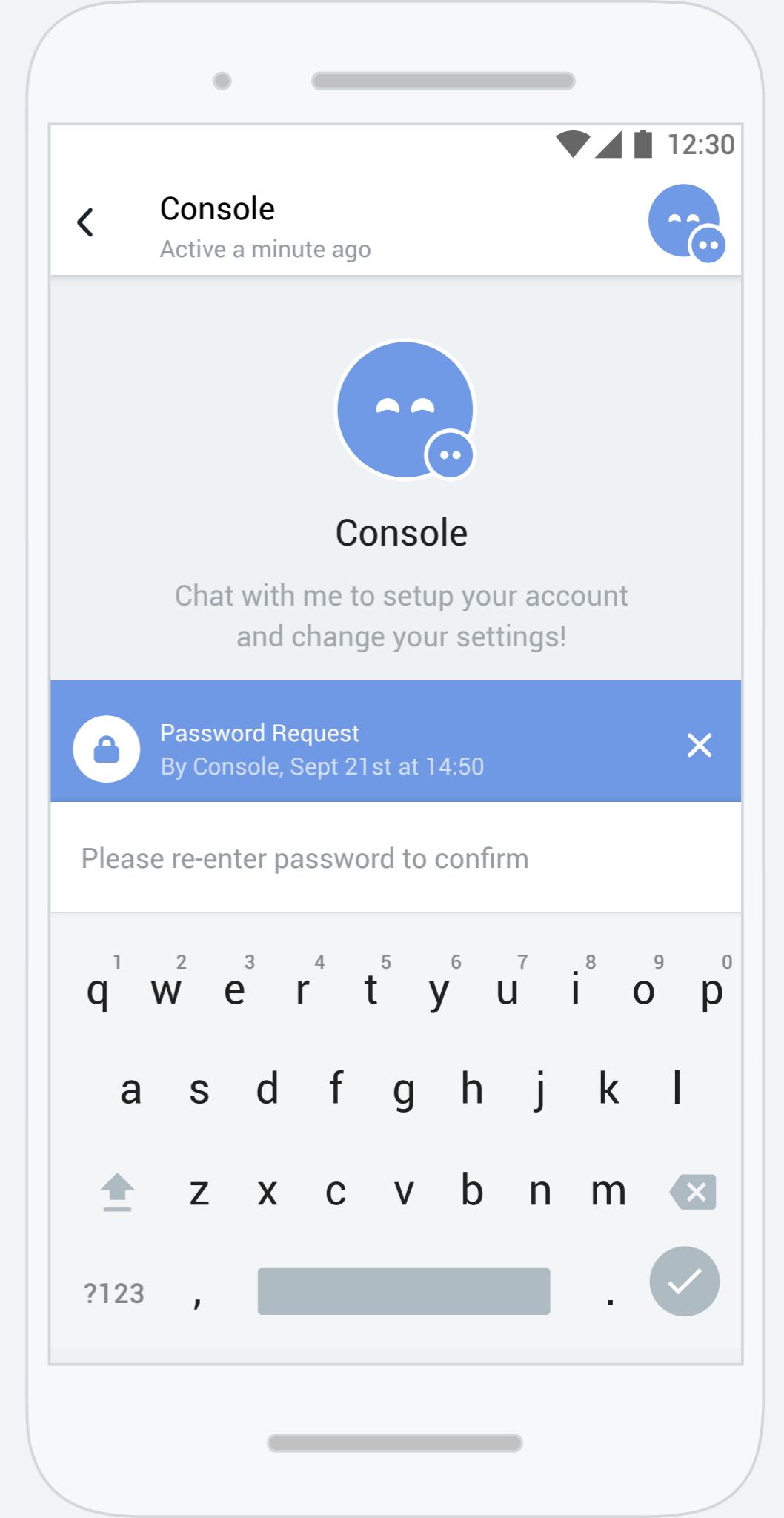
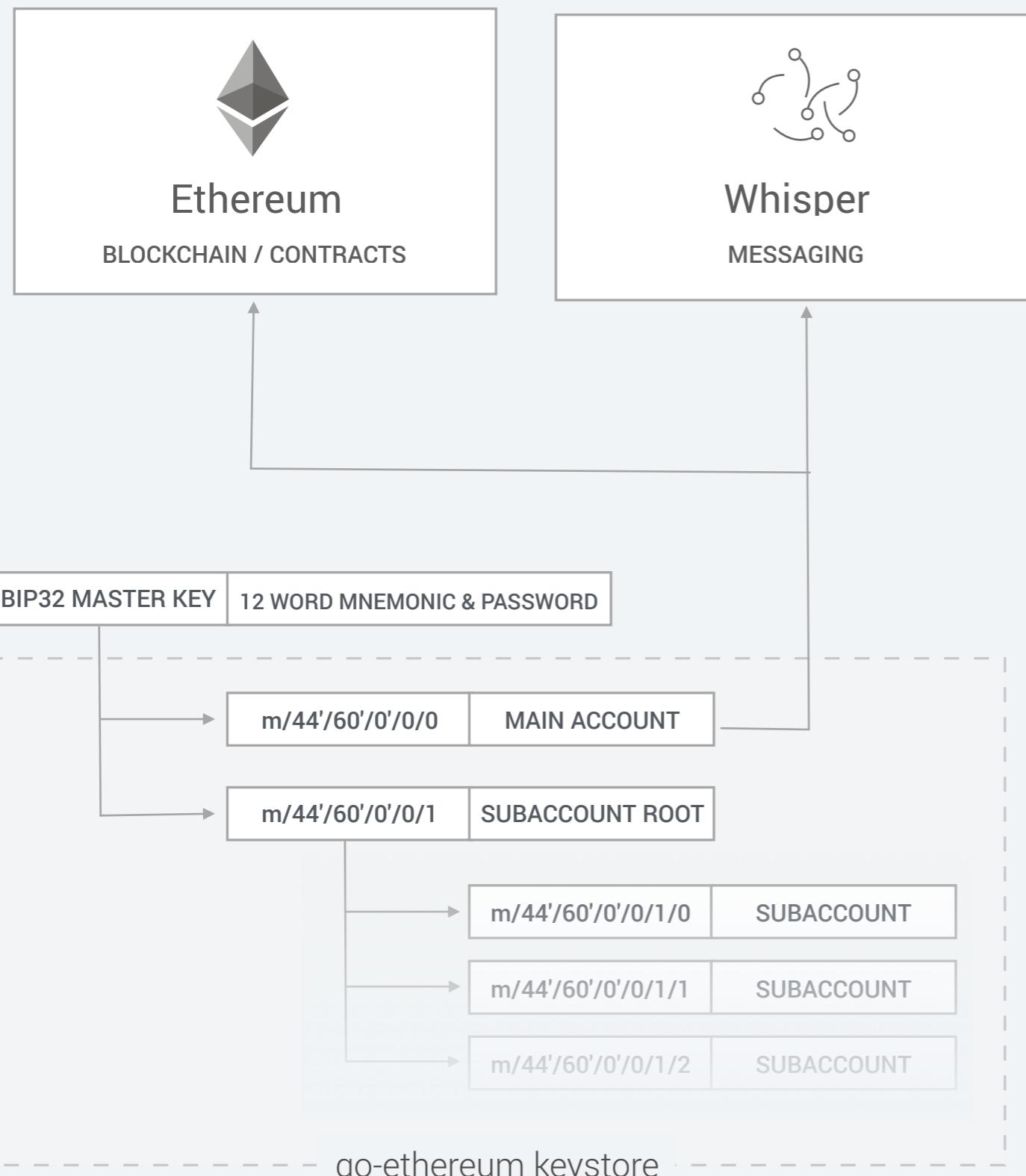
how do we express Ethereum visually?  
what does the DApp ecosystem look like in an IM?

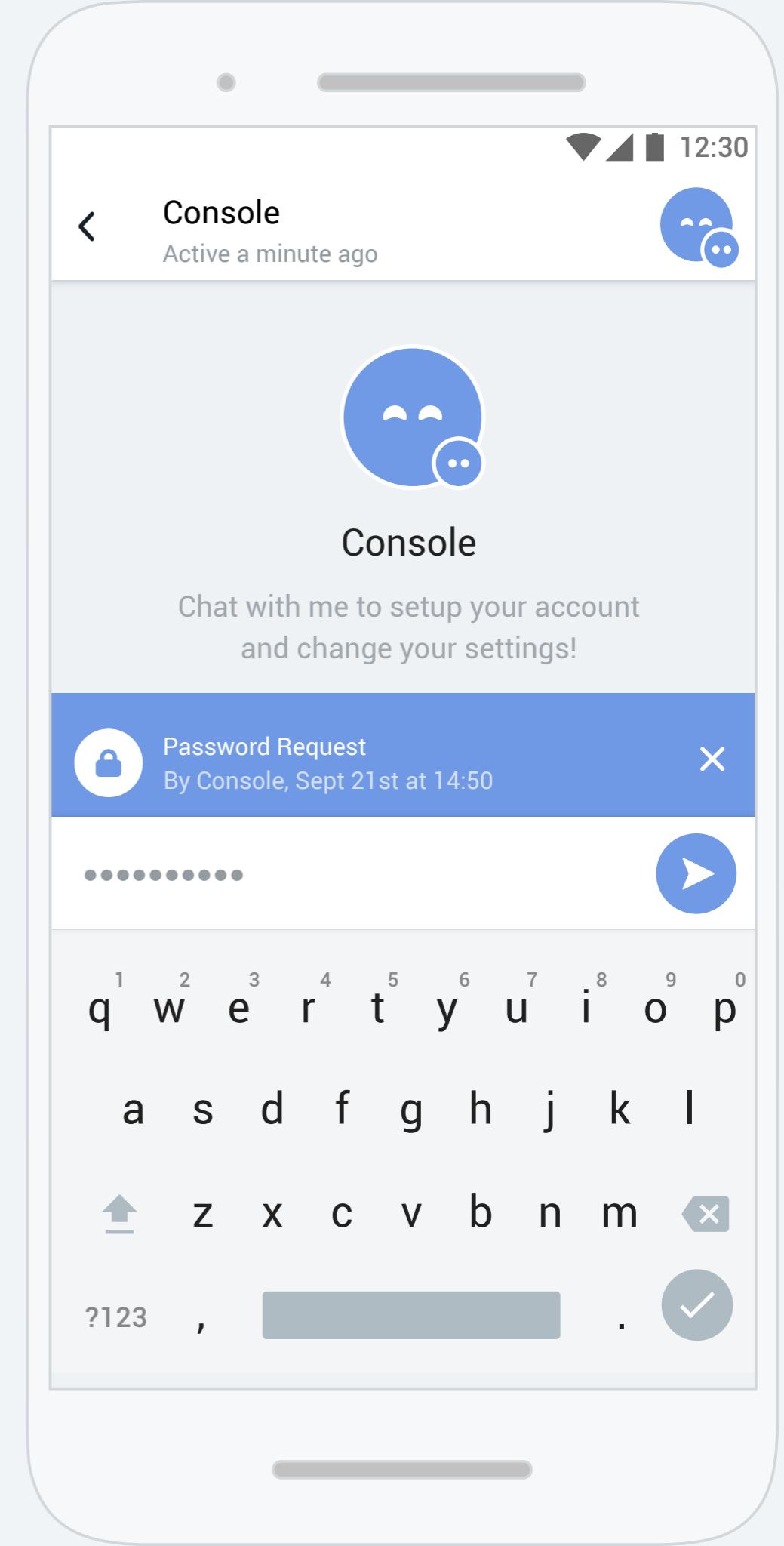
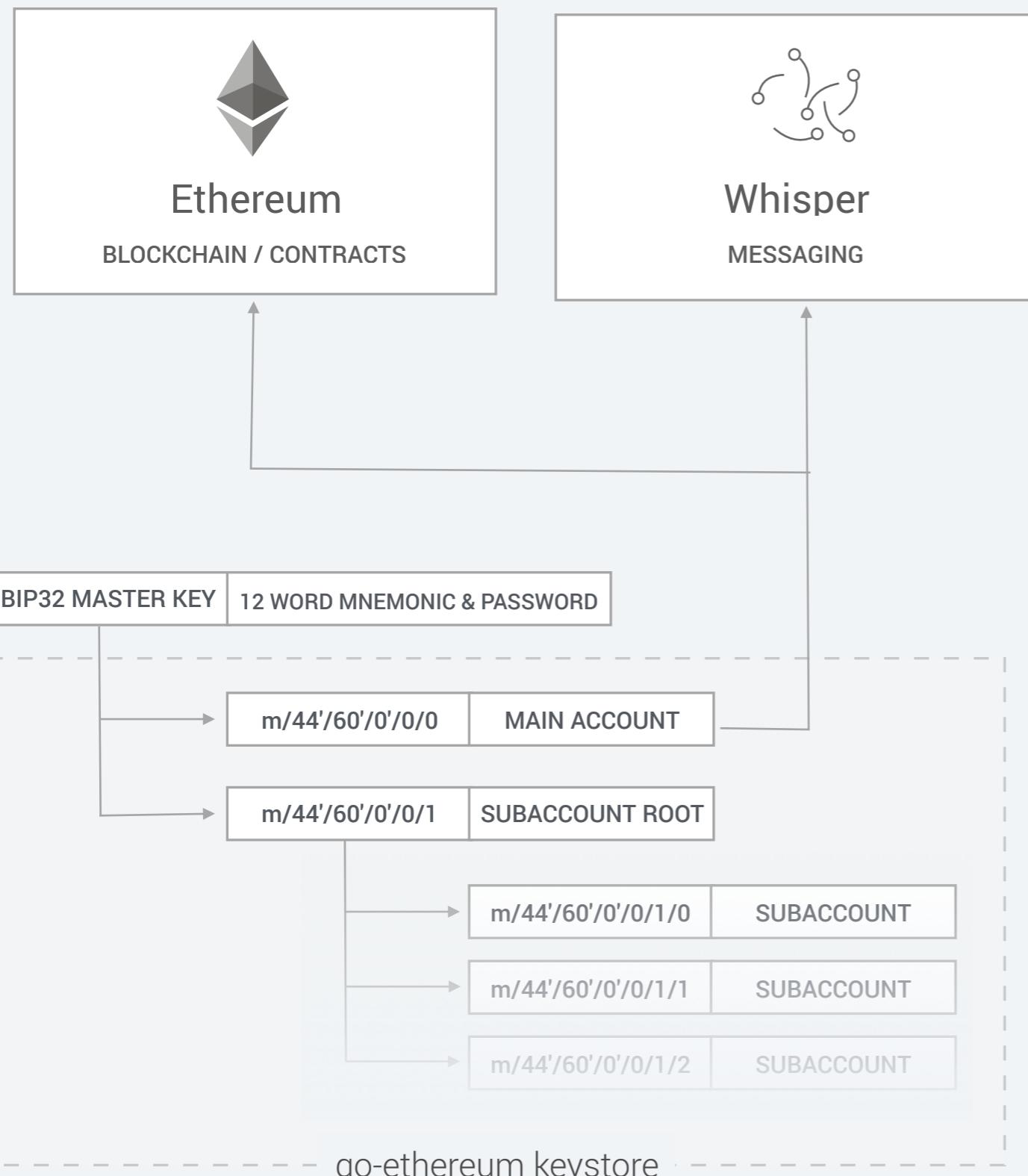


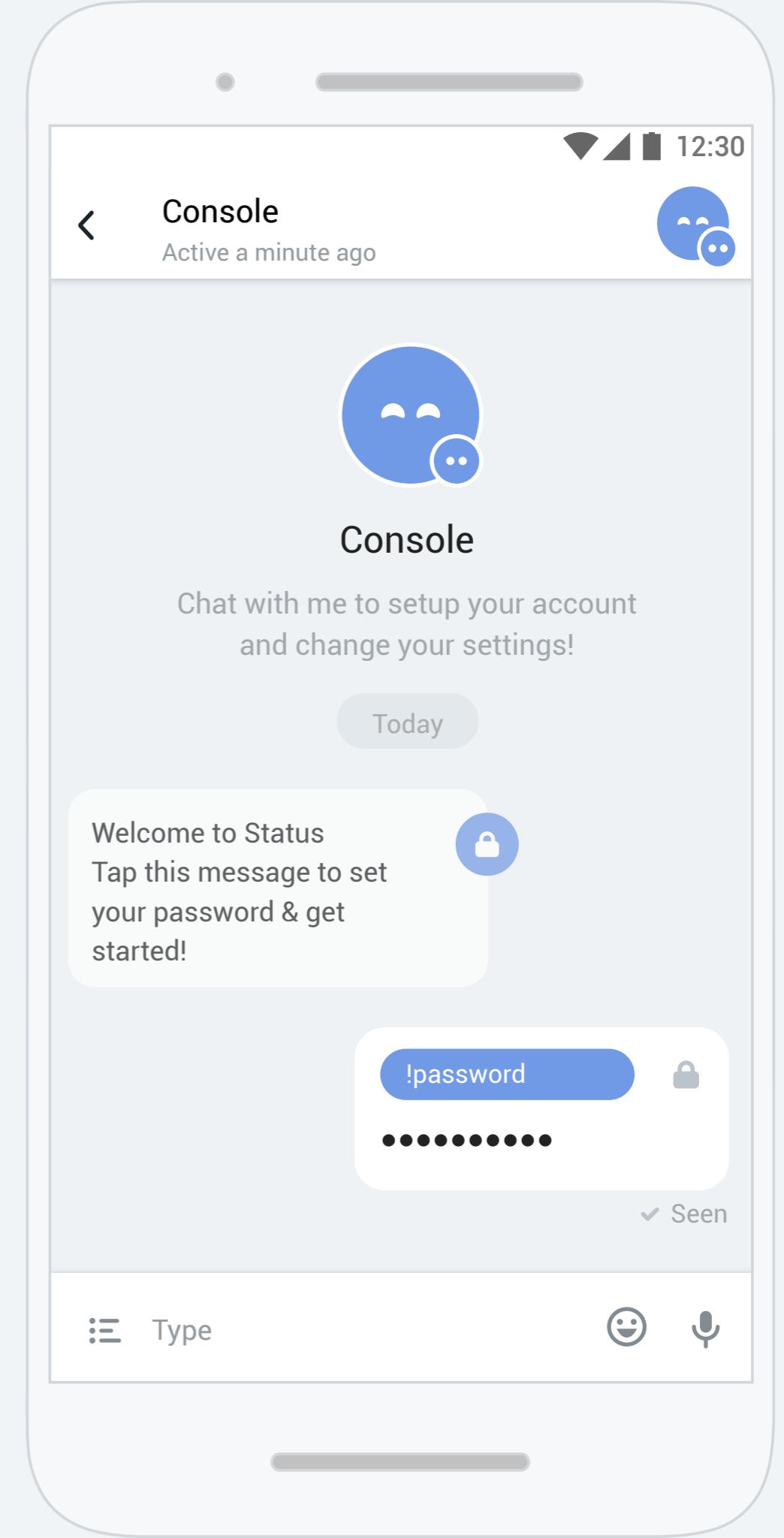
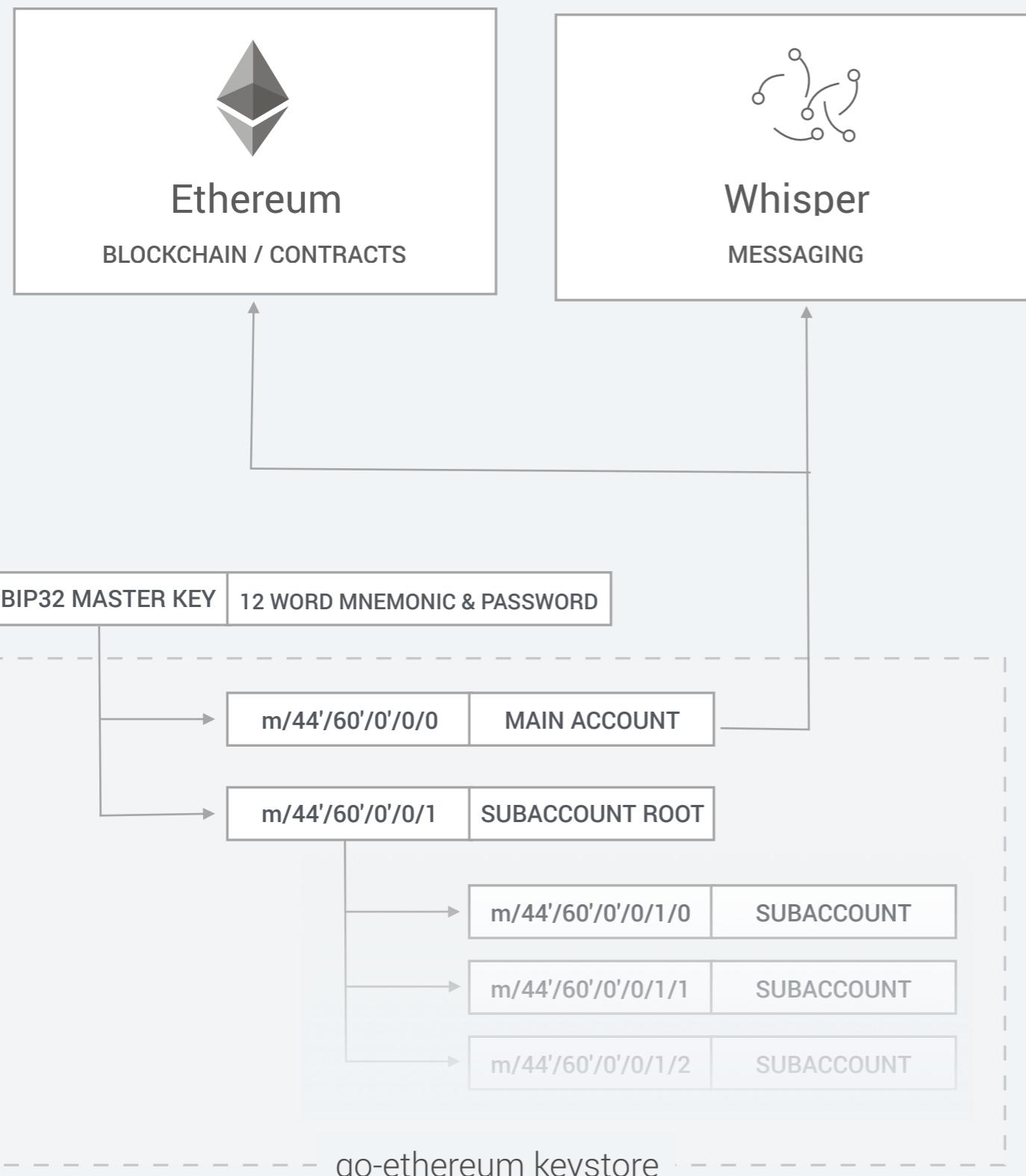


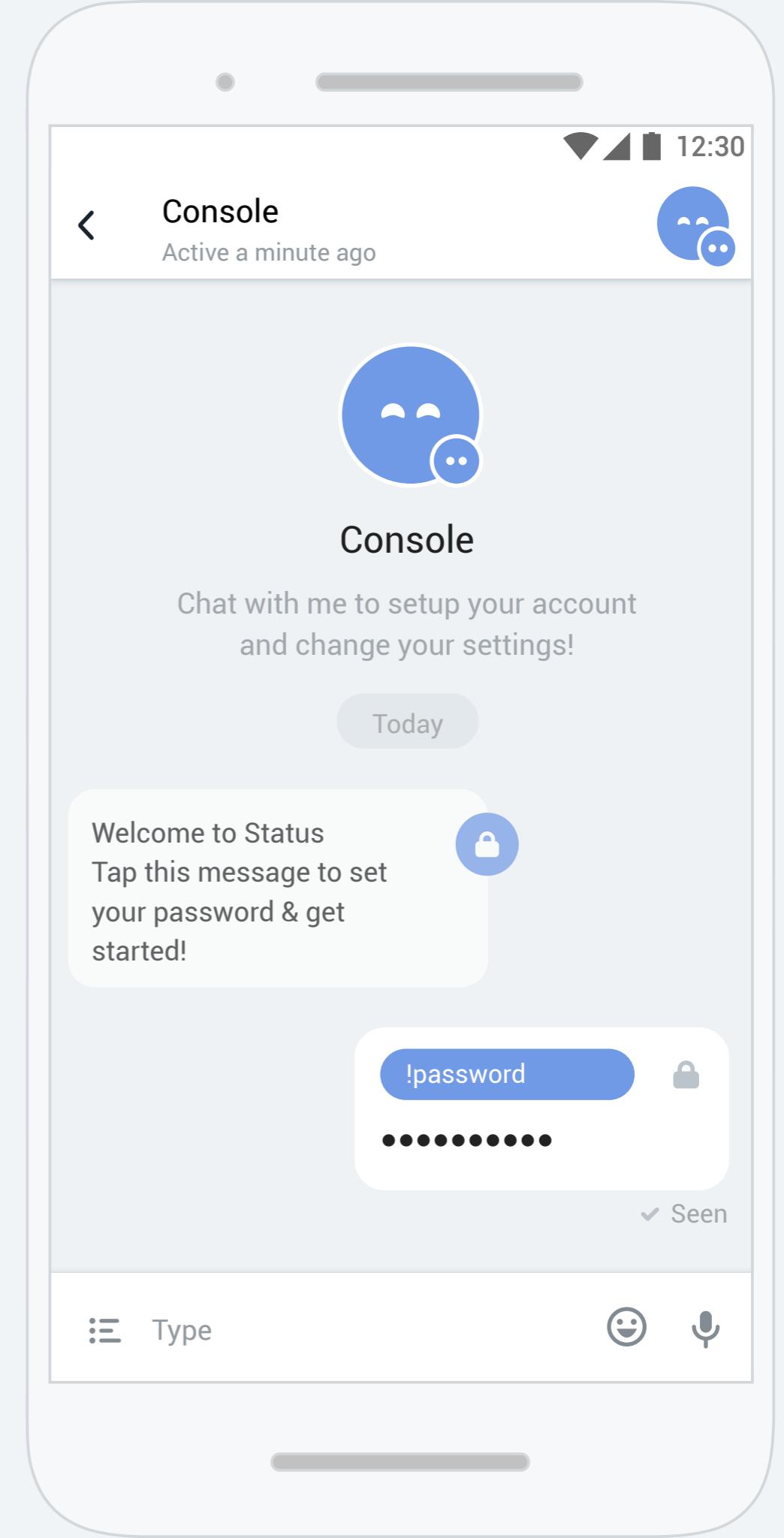
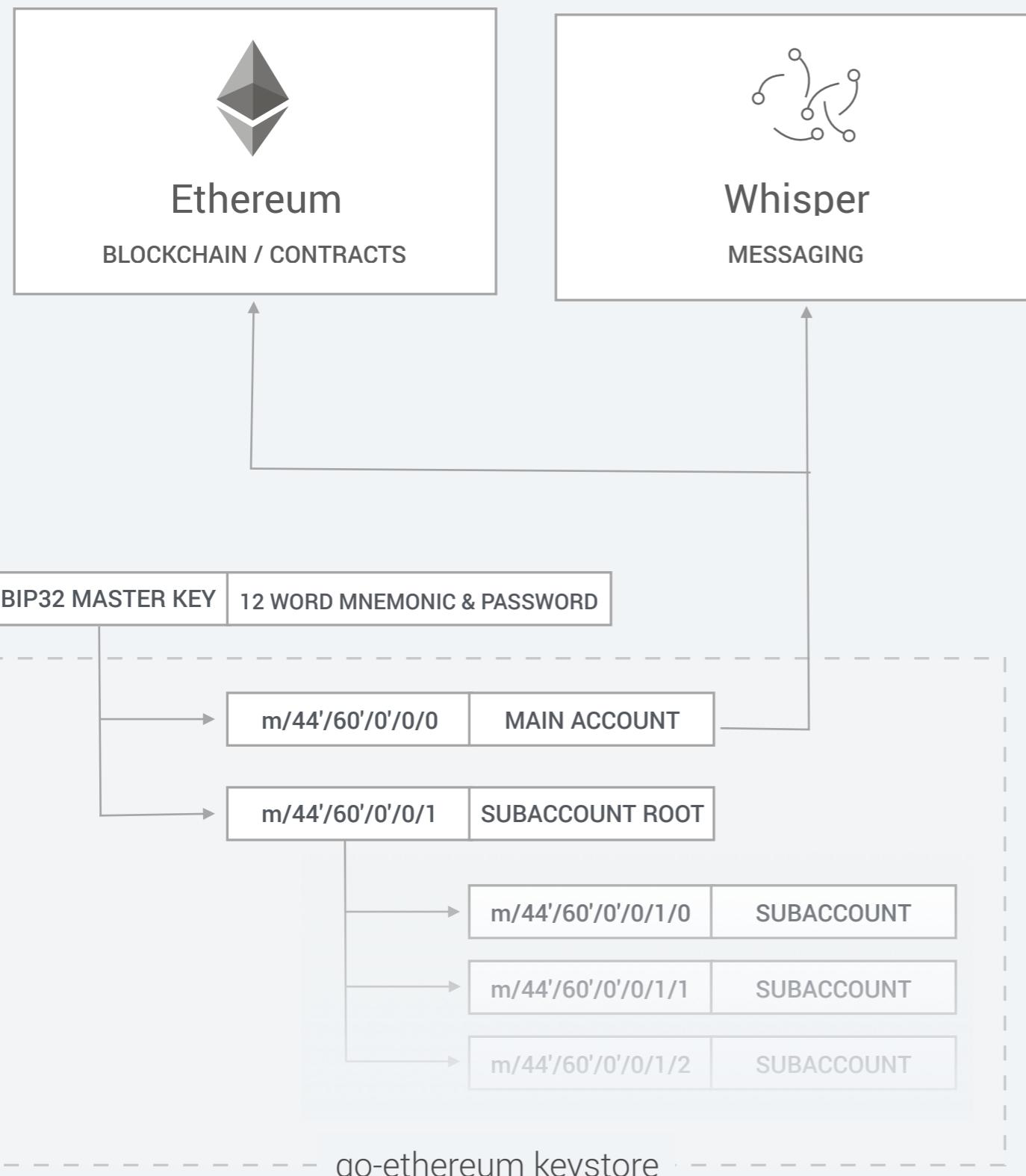


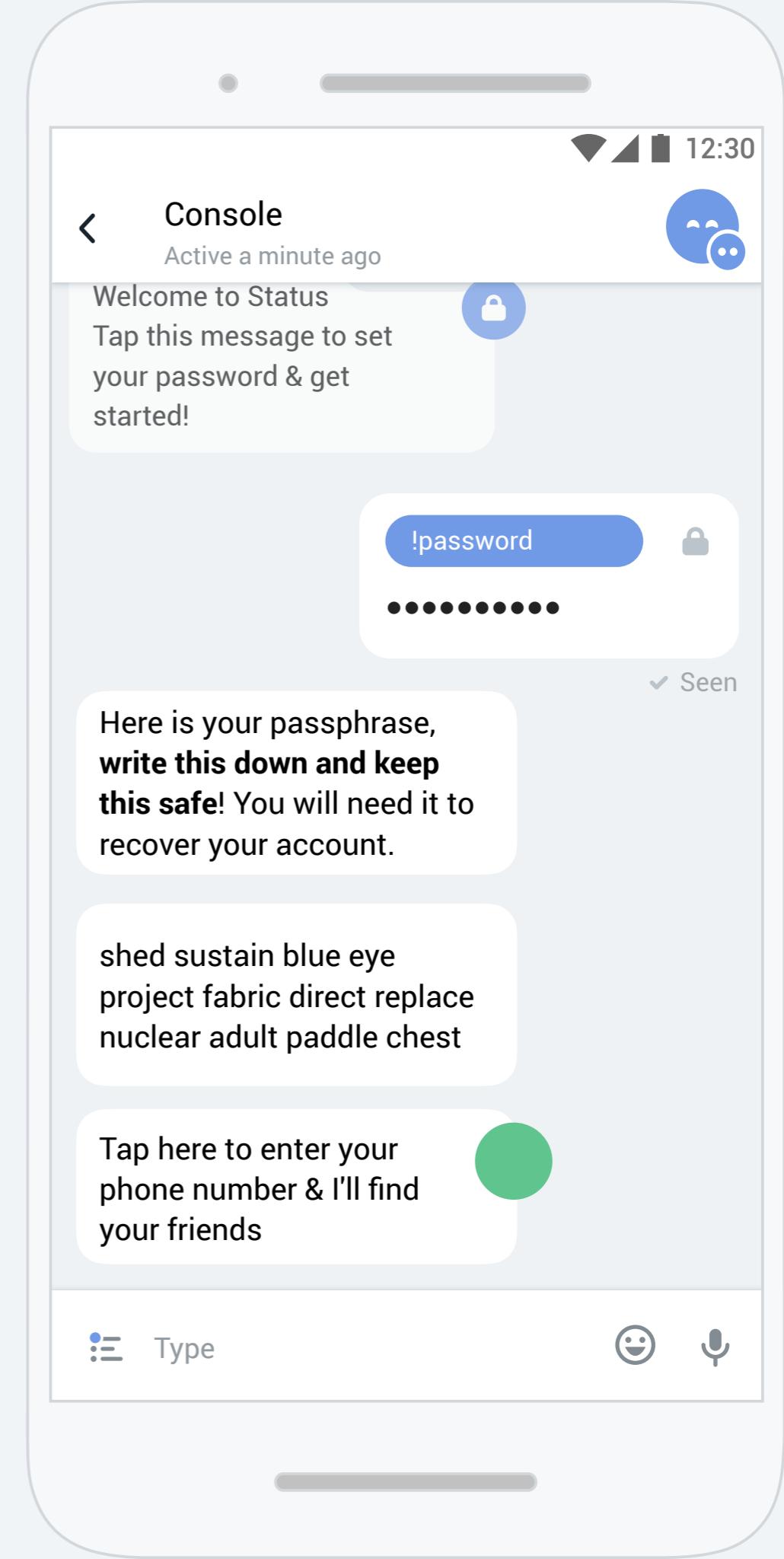
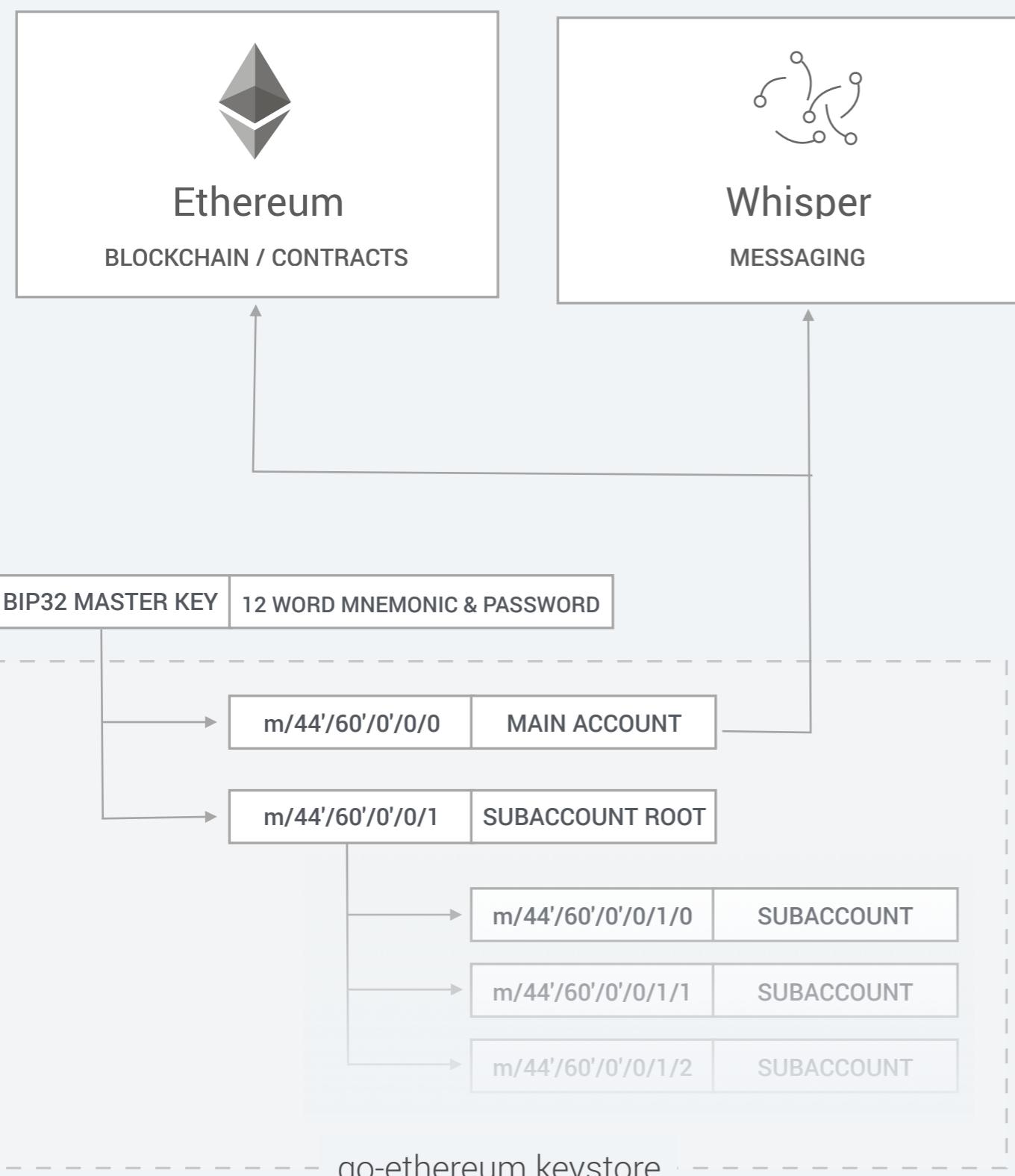




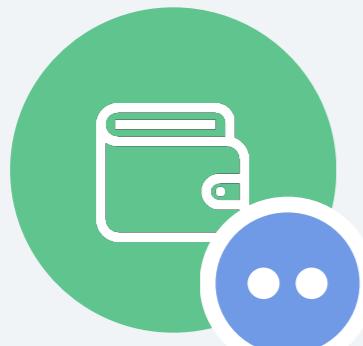




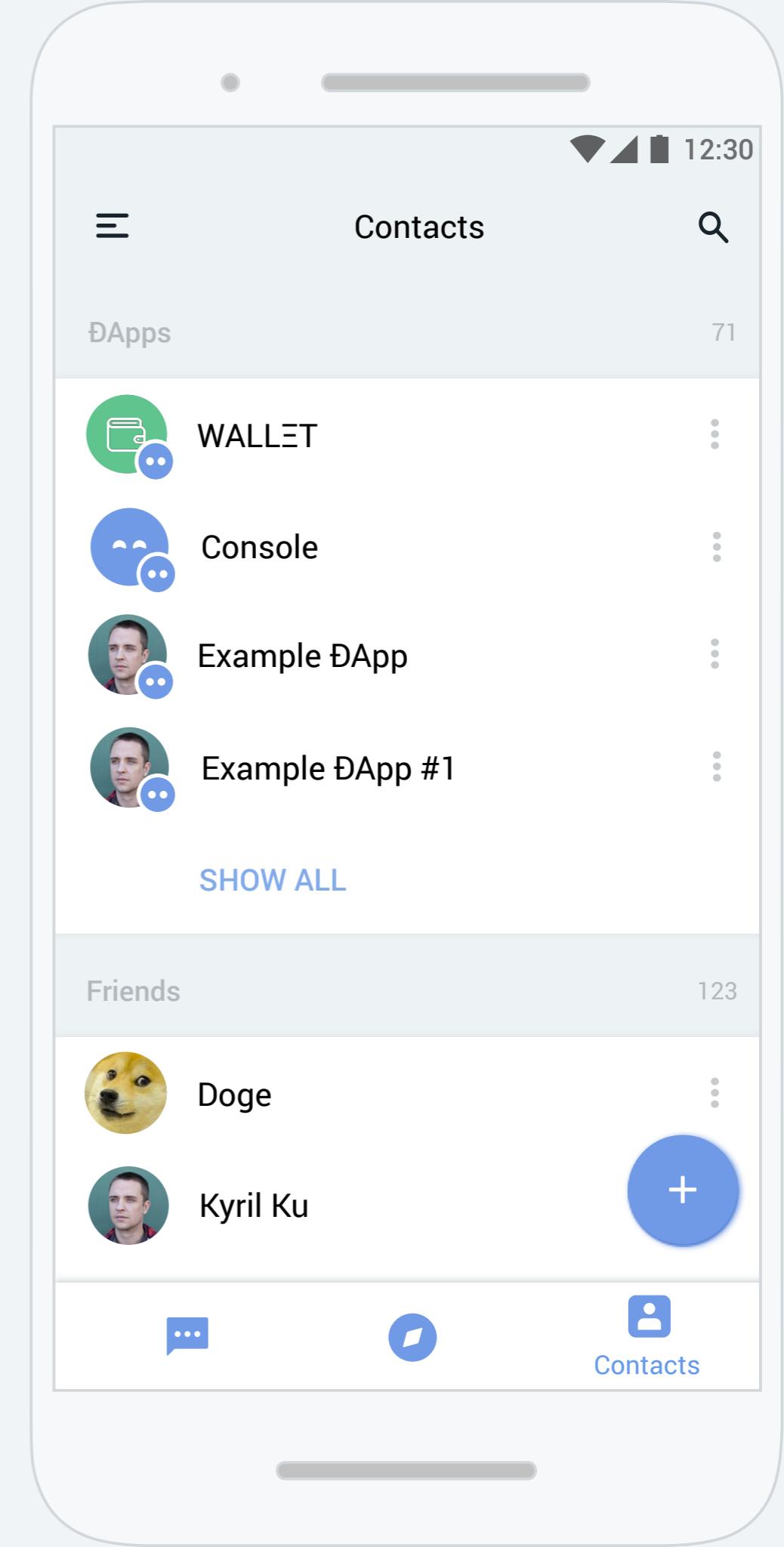




# ÐApps are first class citizens



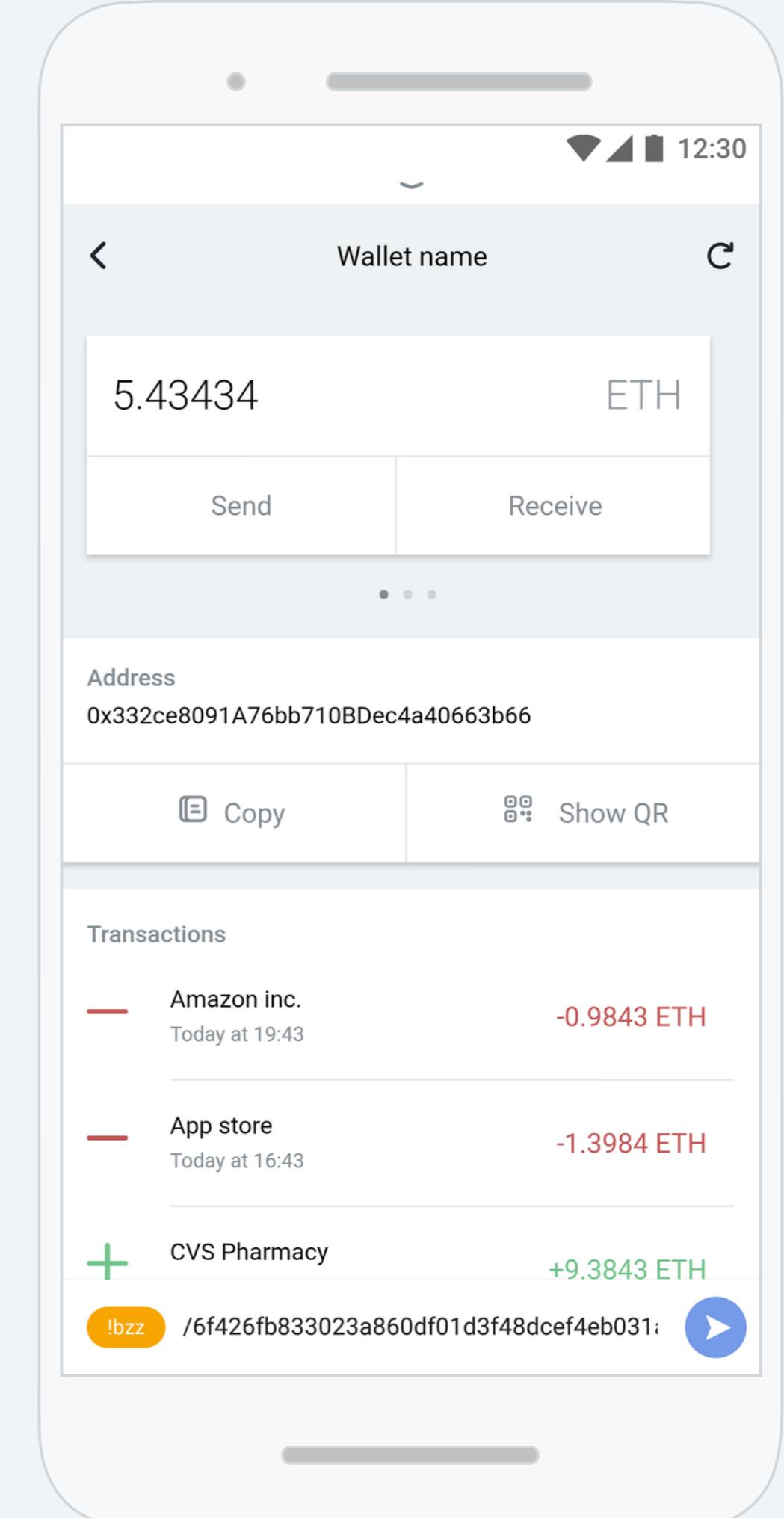
# WALLET



# HTML/JS DApps 🙌

rendered in webkit  
similar to MetaMask & Mist

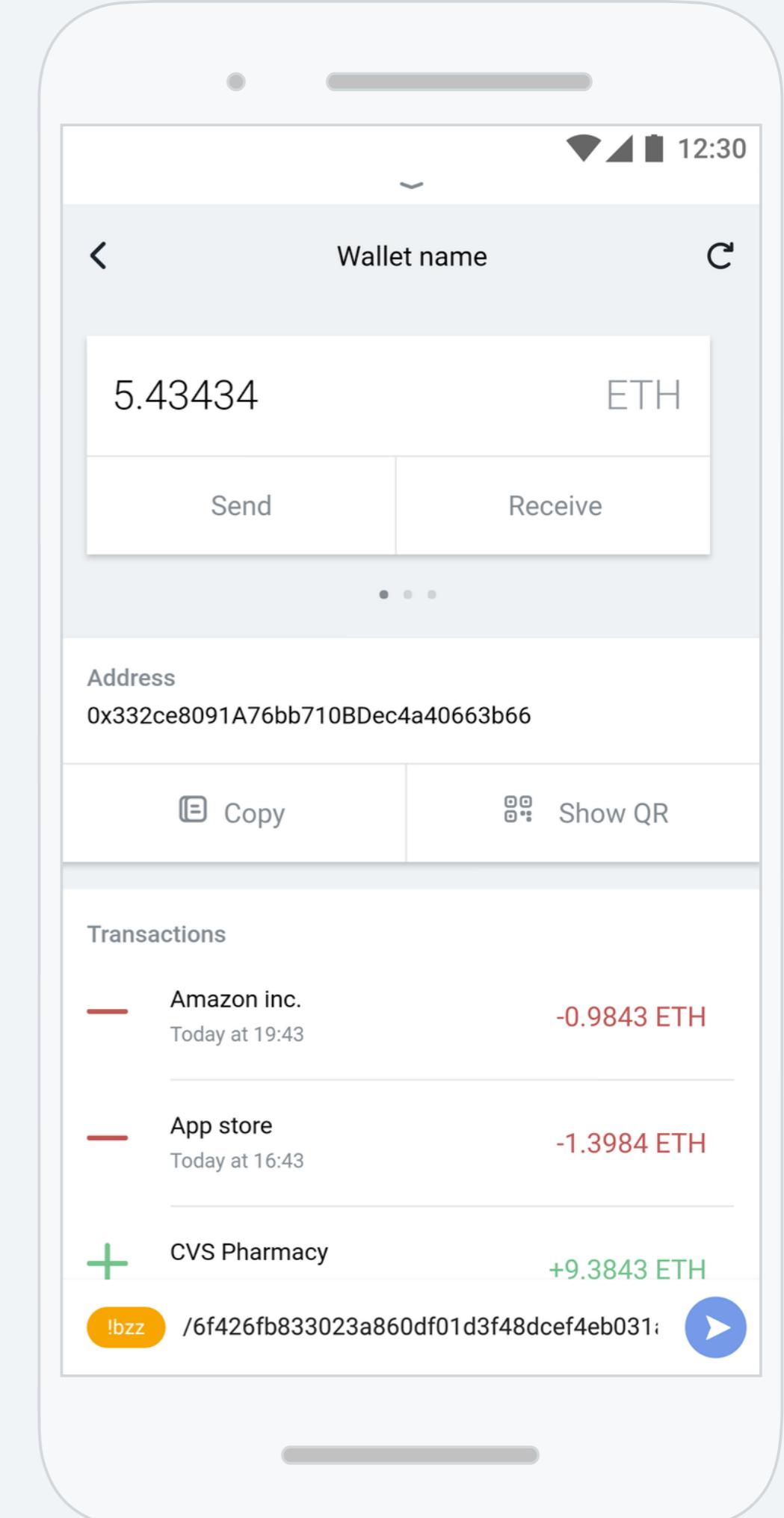
chat history for per DApp transactions  
and can be used as a support channel



# HTML/JS DApps 🙌

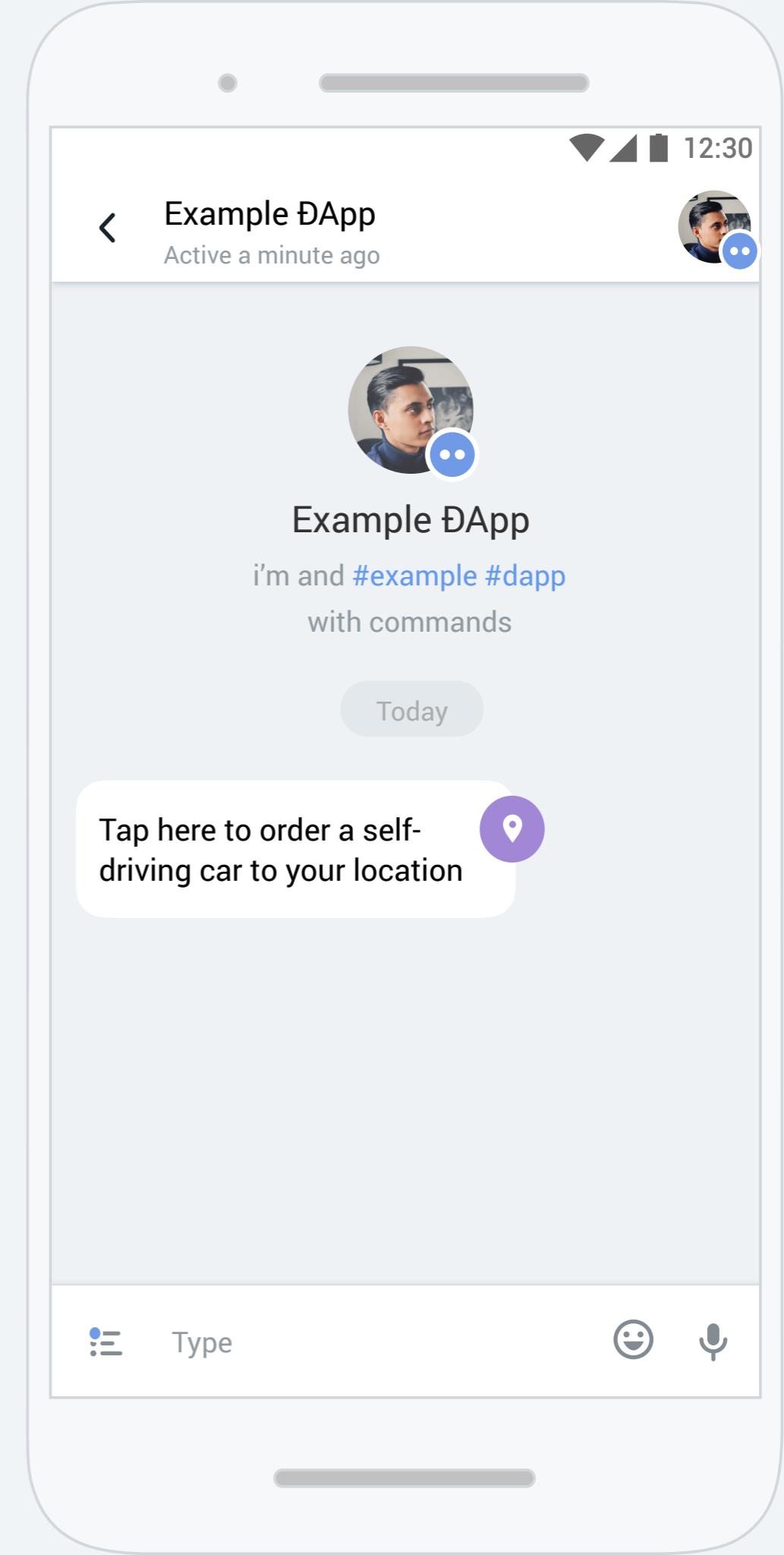
rendered in webkit  
similar to MetaMask & Mist

chat history for per DApp transactions  
and can be used as a support channel



# Chat DApps

uses chat api to;  
create new commands &  
new message types



```

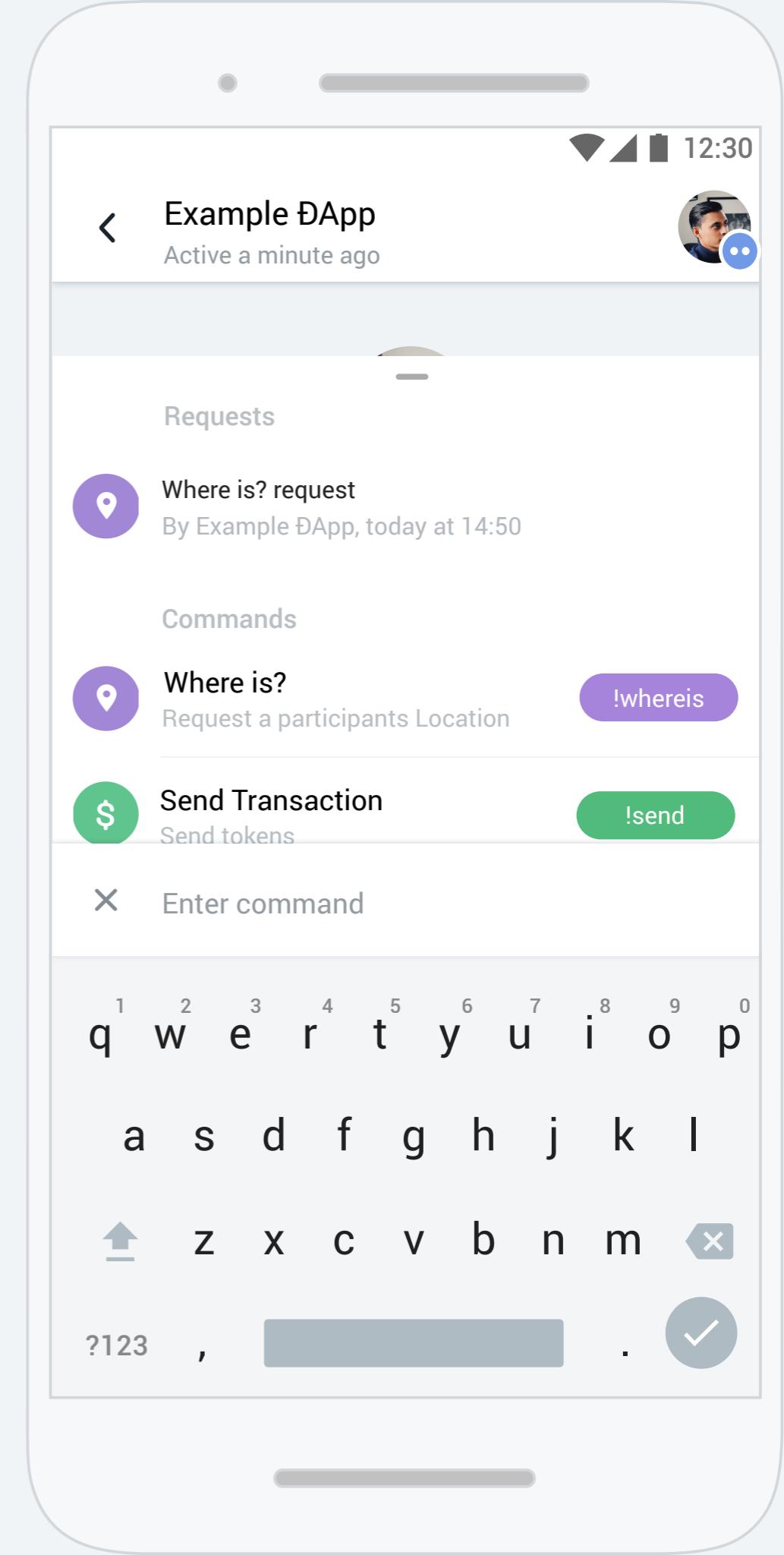
function validateParams(params) {
    // ...
}

function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

    return true;
}

status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});

```



```

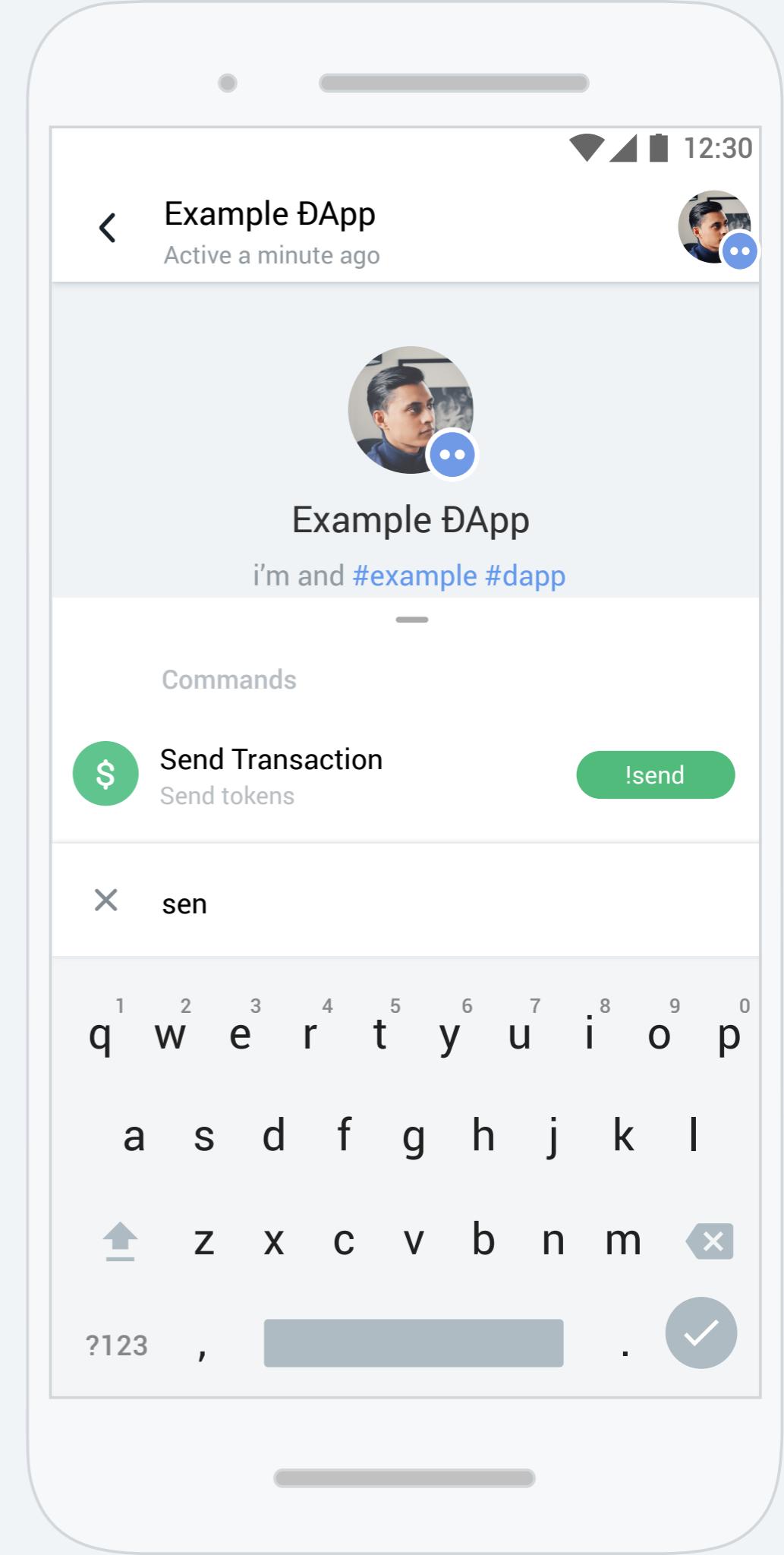
function validateParams(params) {
    // ...
}

function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

    return true;
}

status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});

```



```

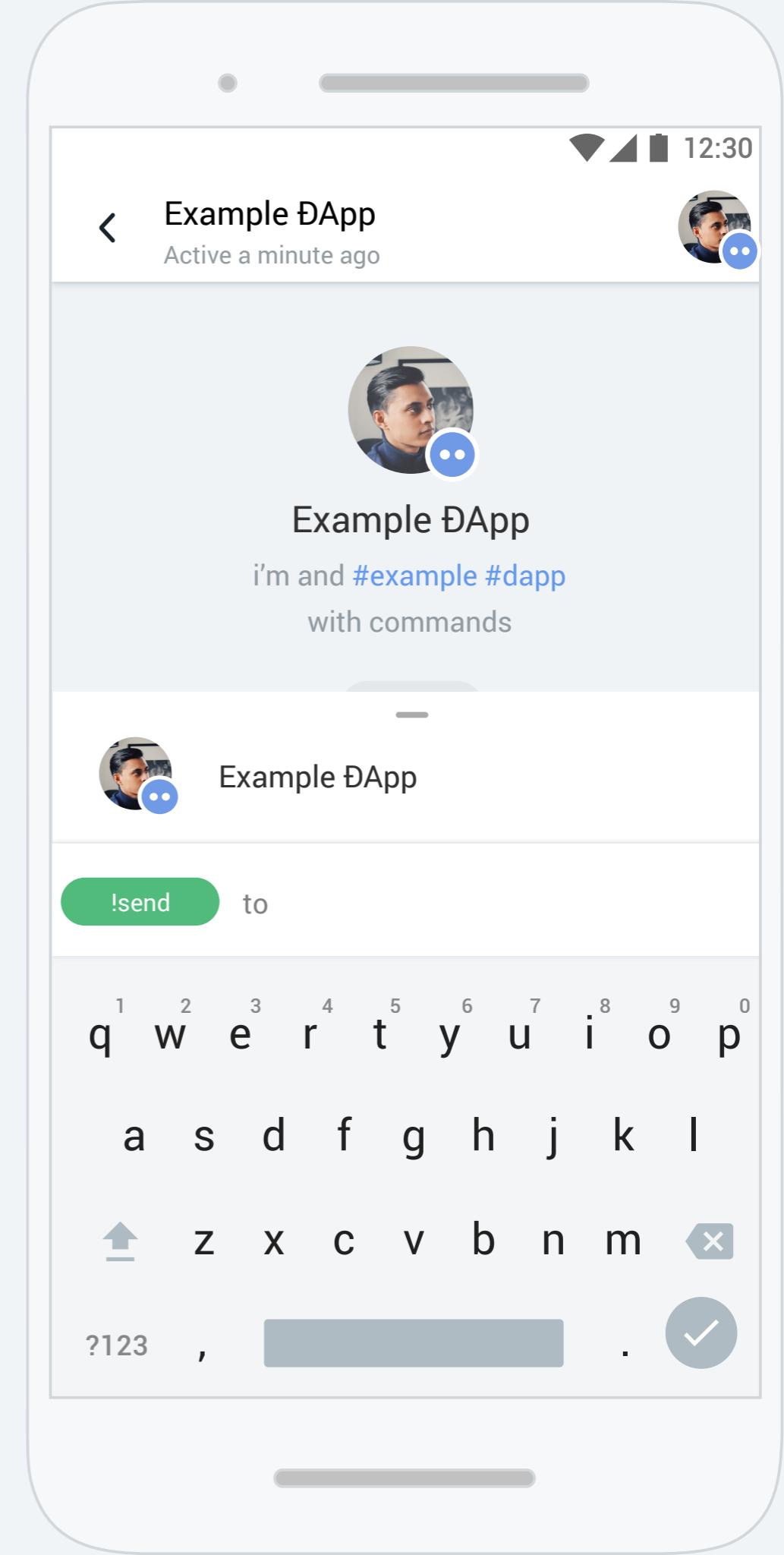
function validateParams(params) {
    // ...
}

function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

    return true;
}

status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});

```

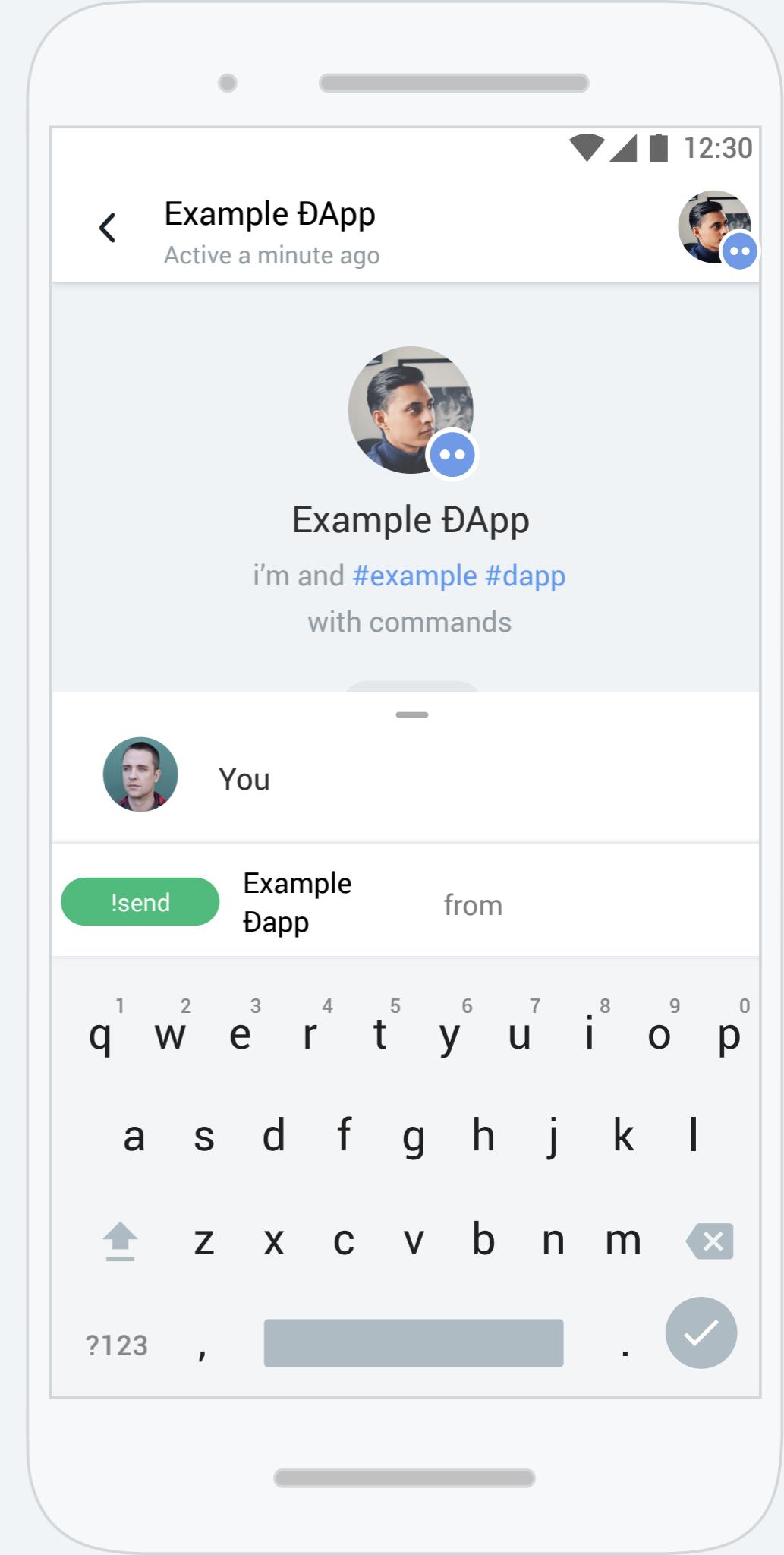


```
function validateParams(params) {
    // ...
}

function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

    return true;
}

status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});
```

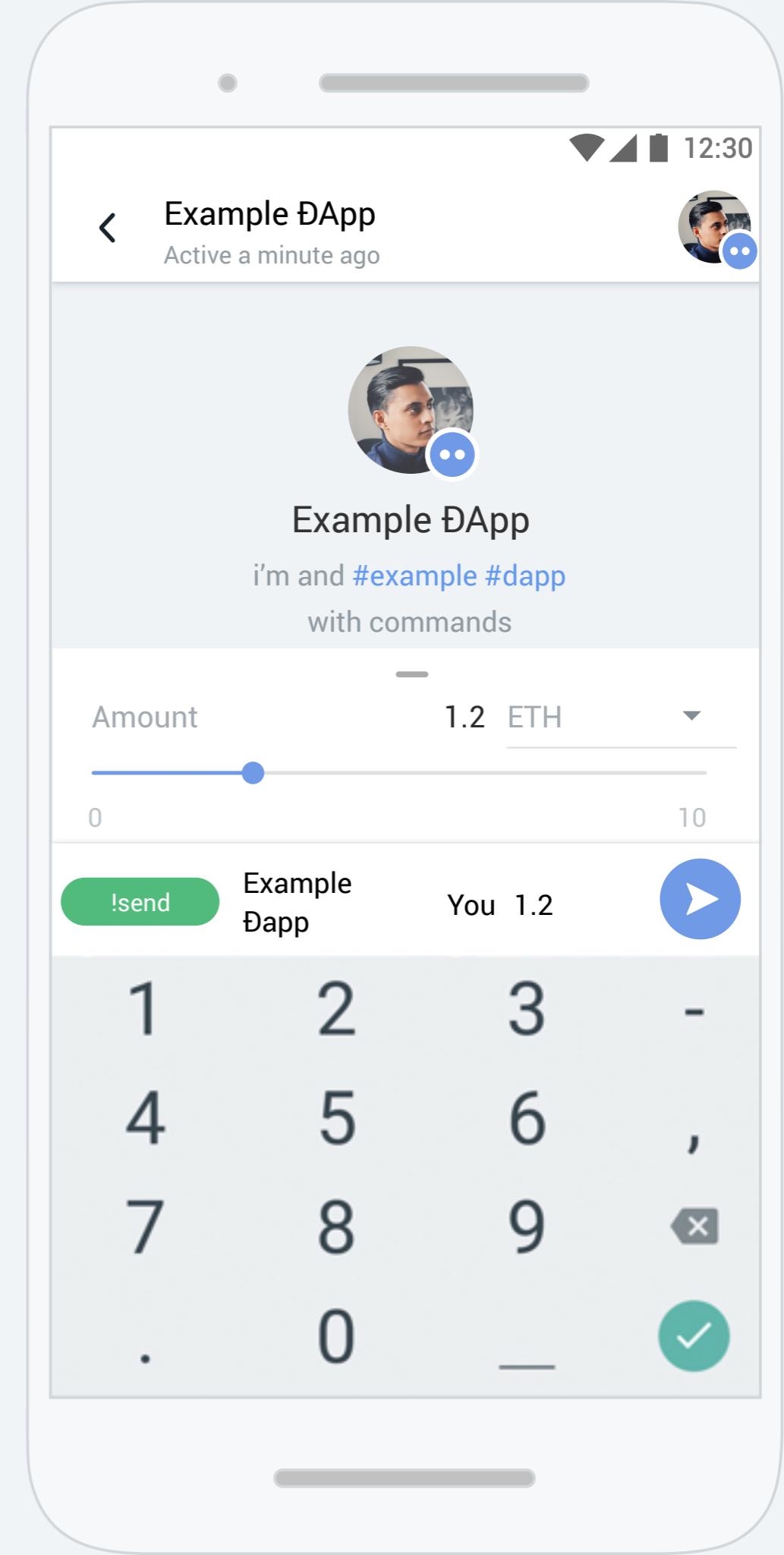


```
function validateParams(params) {
    // ...
}

function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

    return true;
}

status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});
```



```

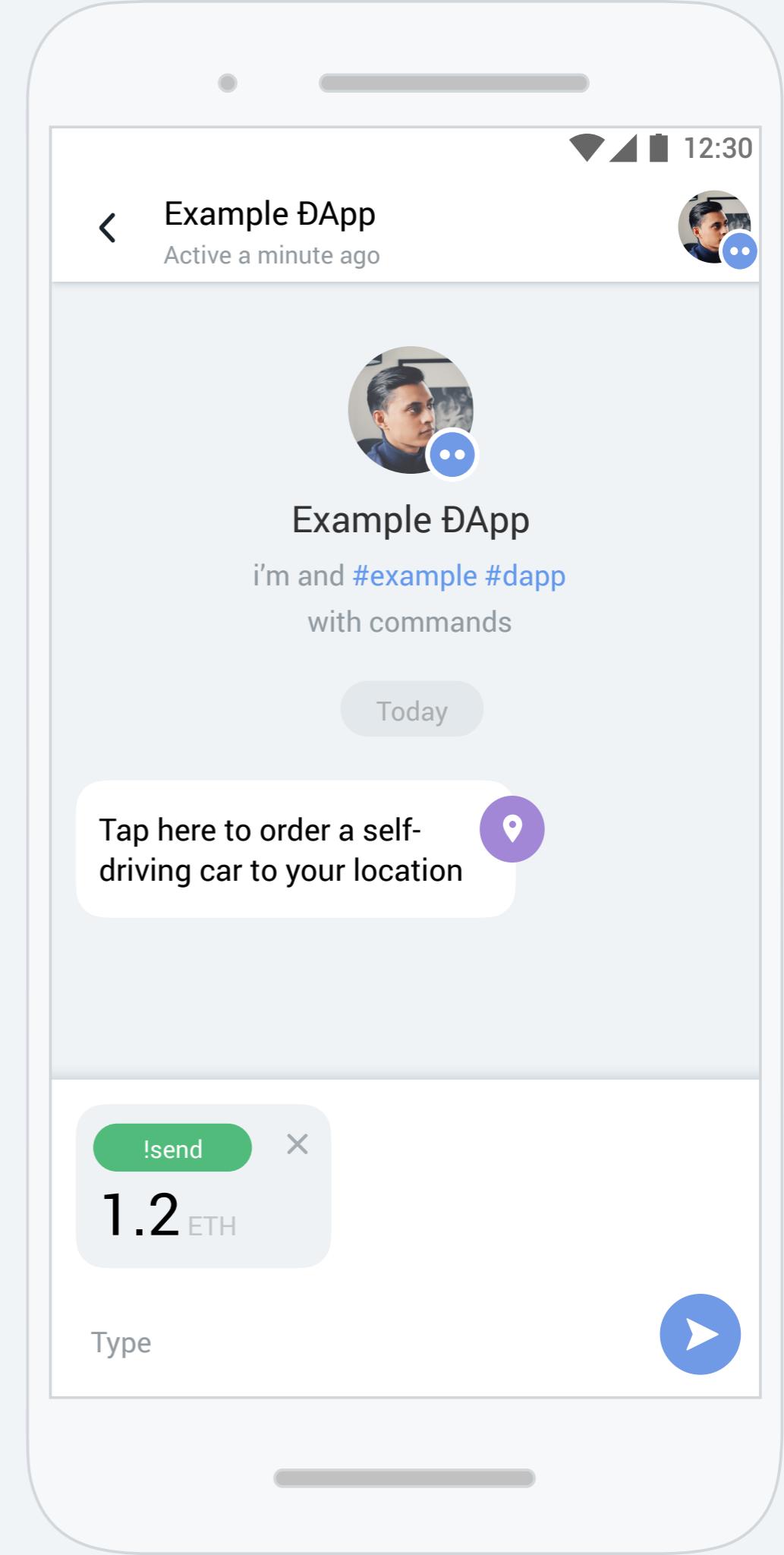
function validateParams(params) {
    // ...
}

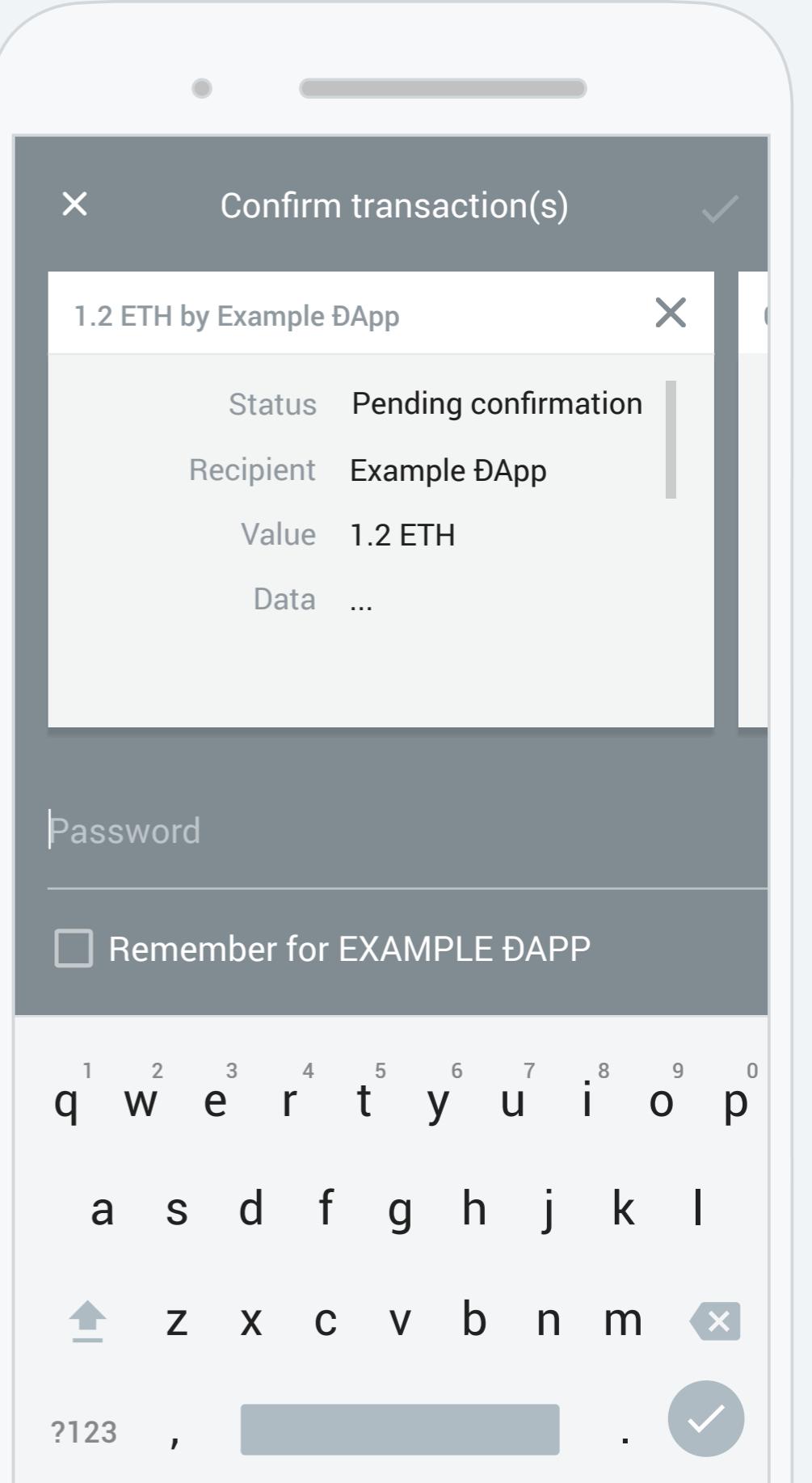
function sendTransaction(params) {
    var data = {
        from: params.from,
        to: params.to,
        value: web3.toWei(params.amount, "ether")
    };
    var hash = web3.eth.sendTransaction(data);

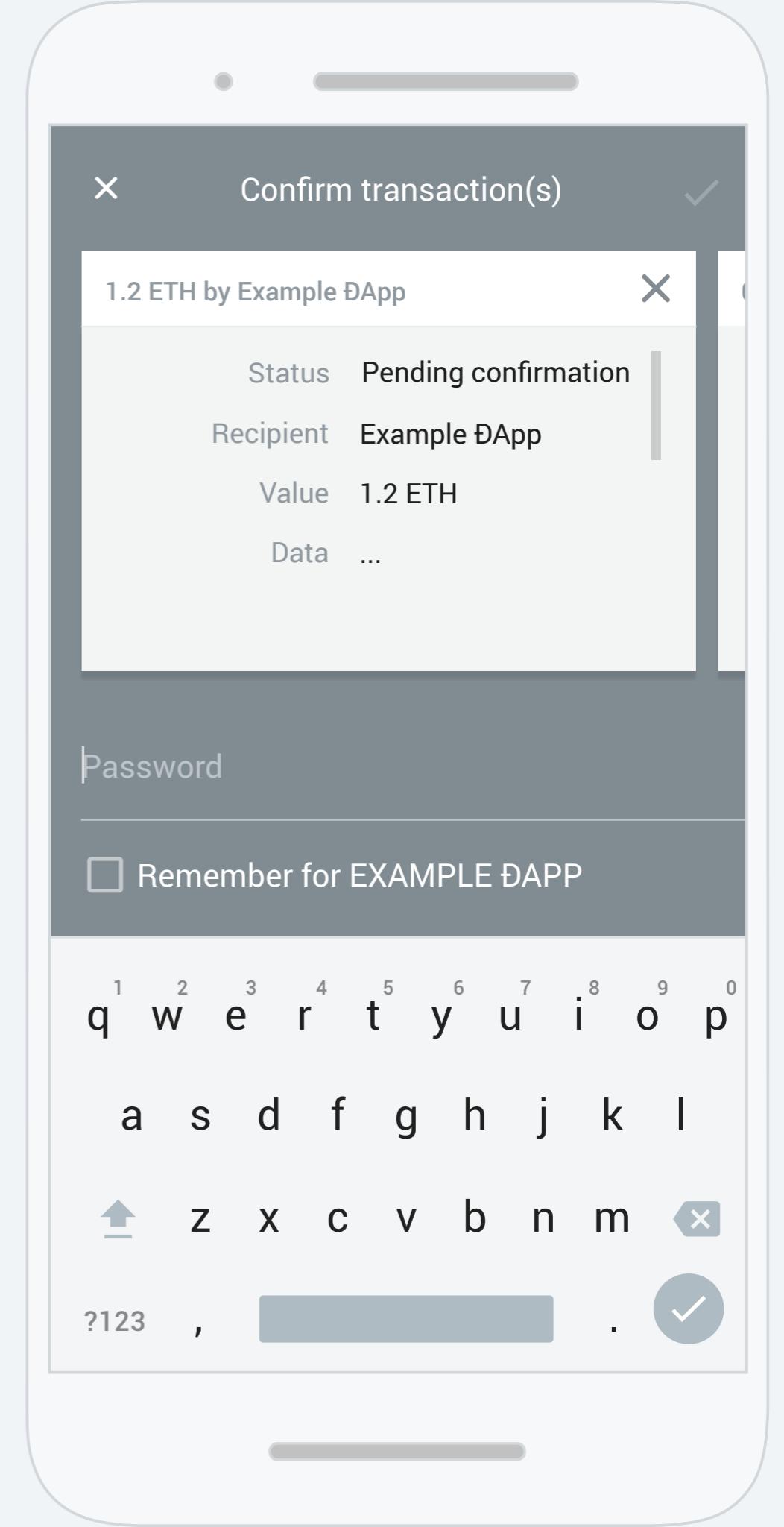
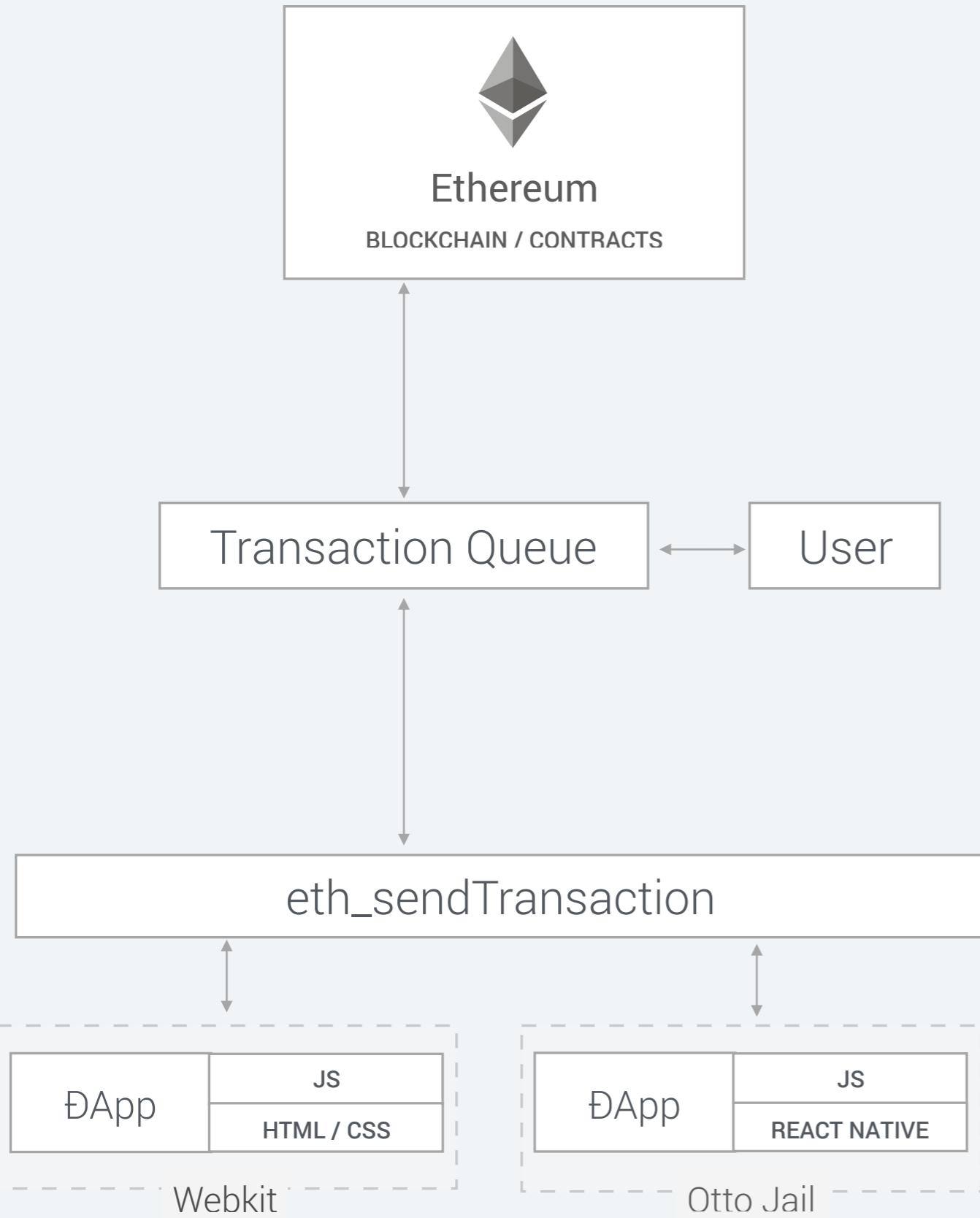
    return true;
}

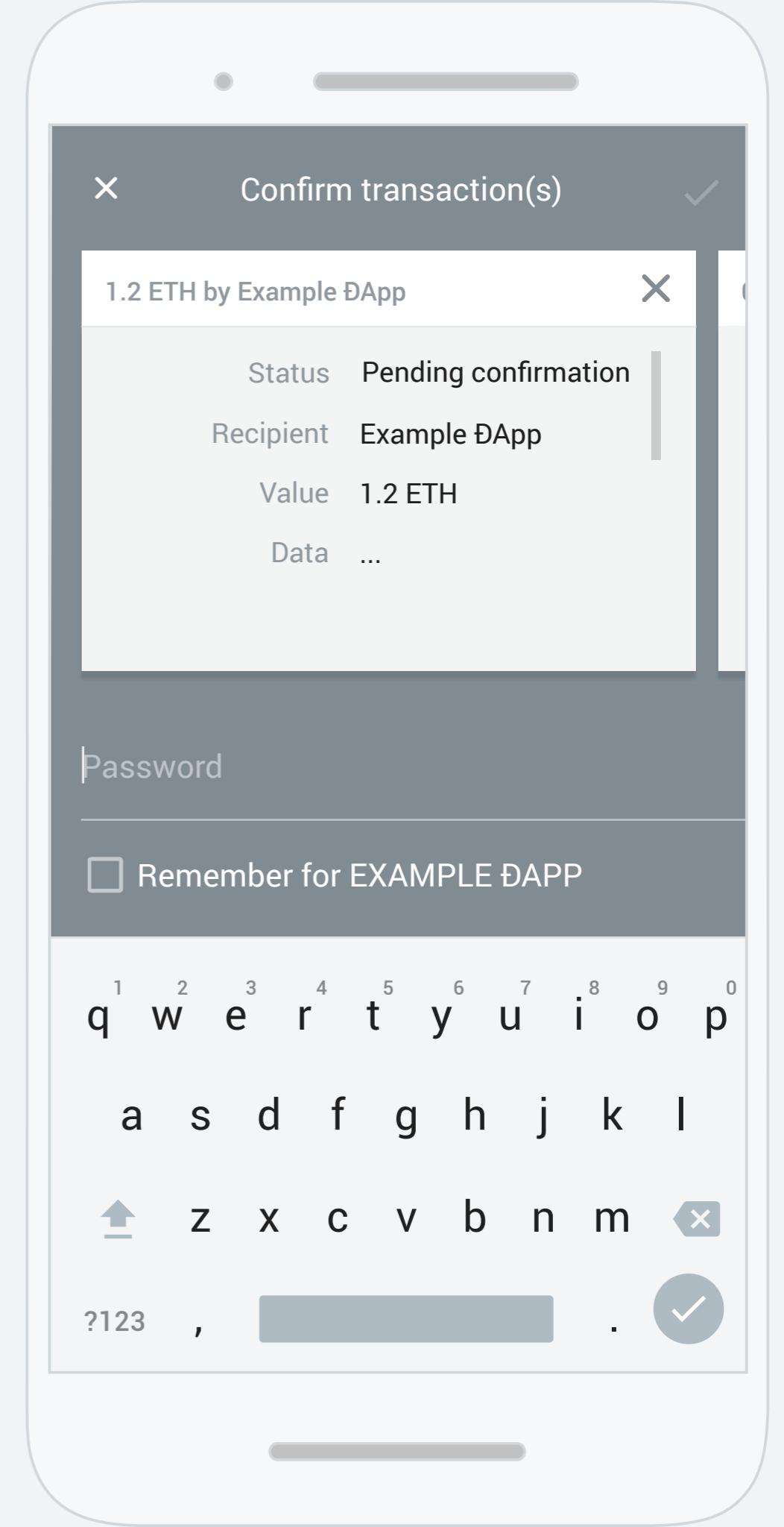
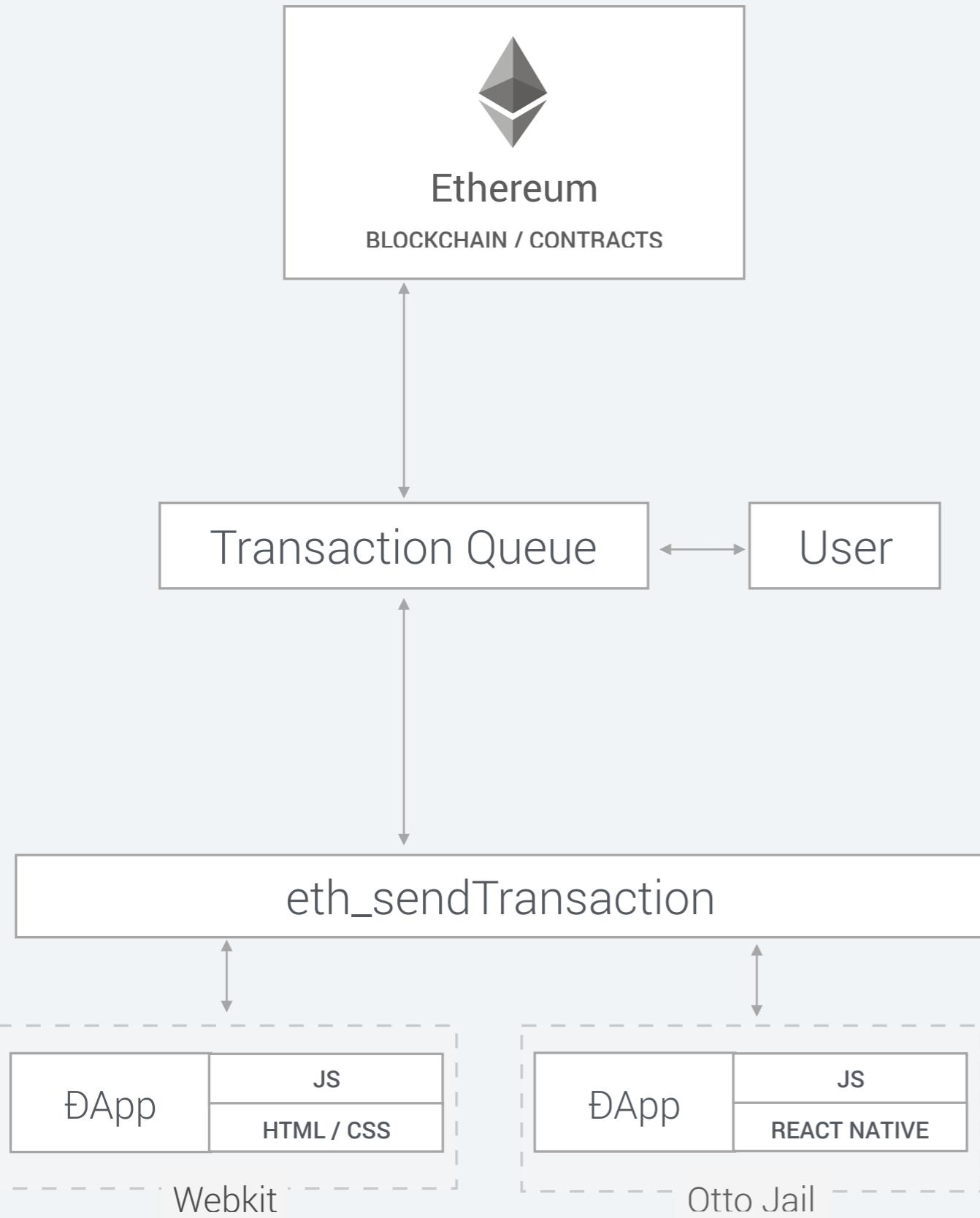
status.command({
    name: "send",
    color: "#50bb7a",
    description: "Send transaction",
    params: [
        {
            name: "to",
            type: status.types.CONTACT,
            default: context.participants[0]
        }, {
            name: "from",
            type: status.types.CONTACT,
            default: context.self
        }, {
            name: "amount",
            type: status.types.NUMBER
        }
    ],
    preview: function (params) {
        return status.components.text(
            {},
            params.value + " ETH"
        );
    },
    handler: sendTransaction,
    validator: validateParams
});

```





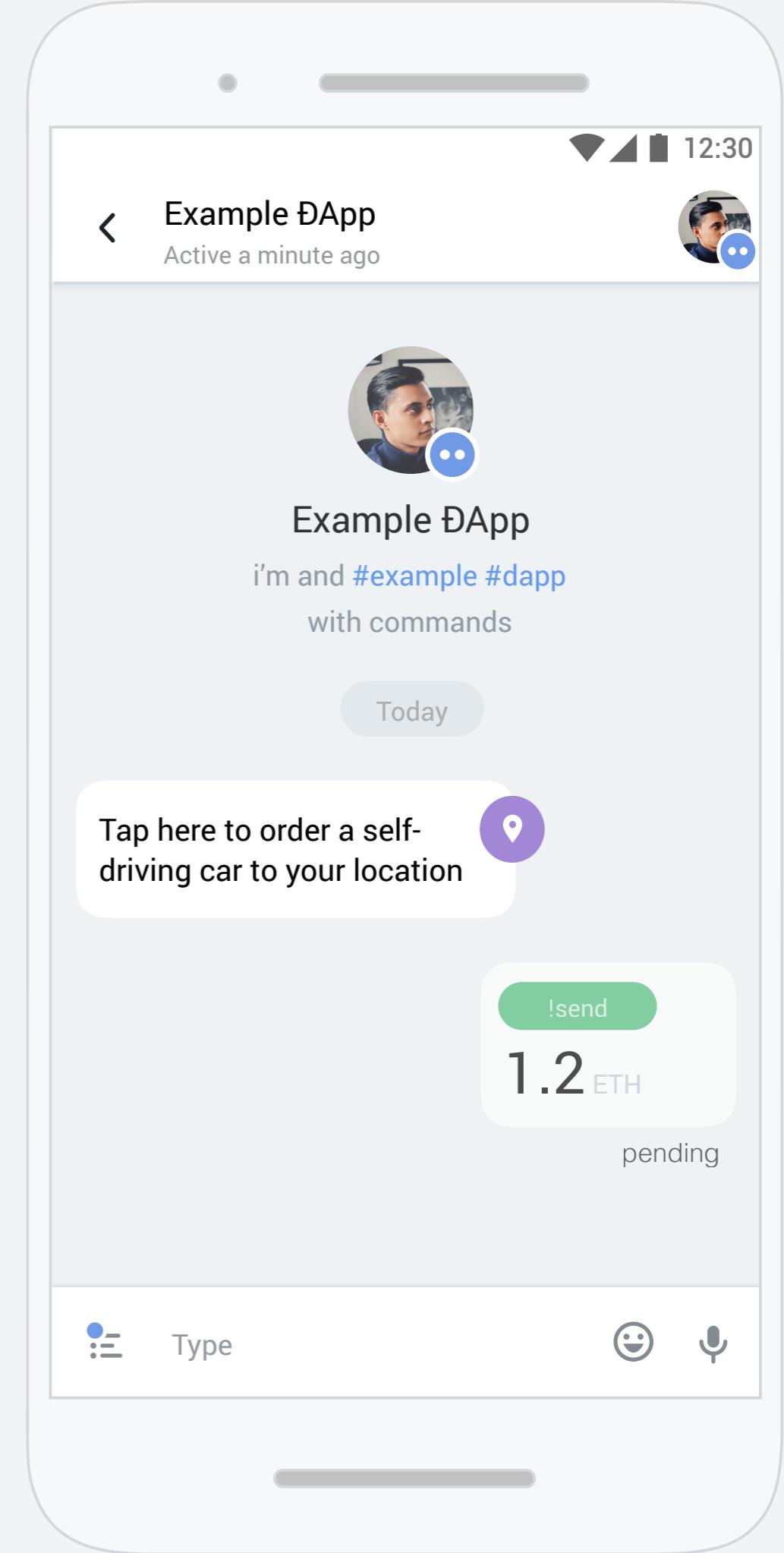




# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

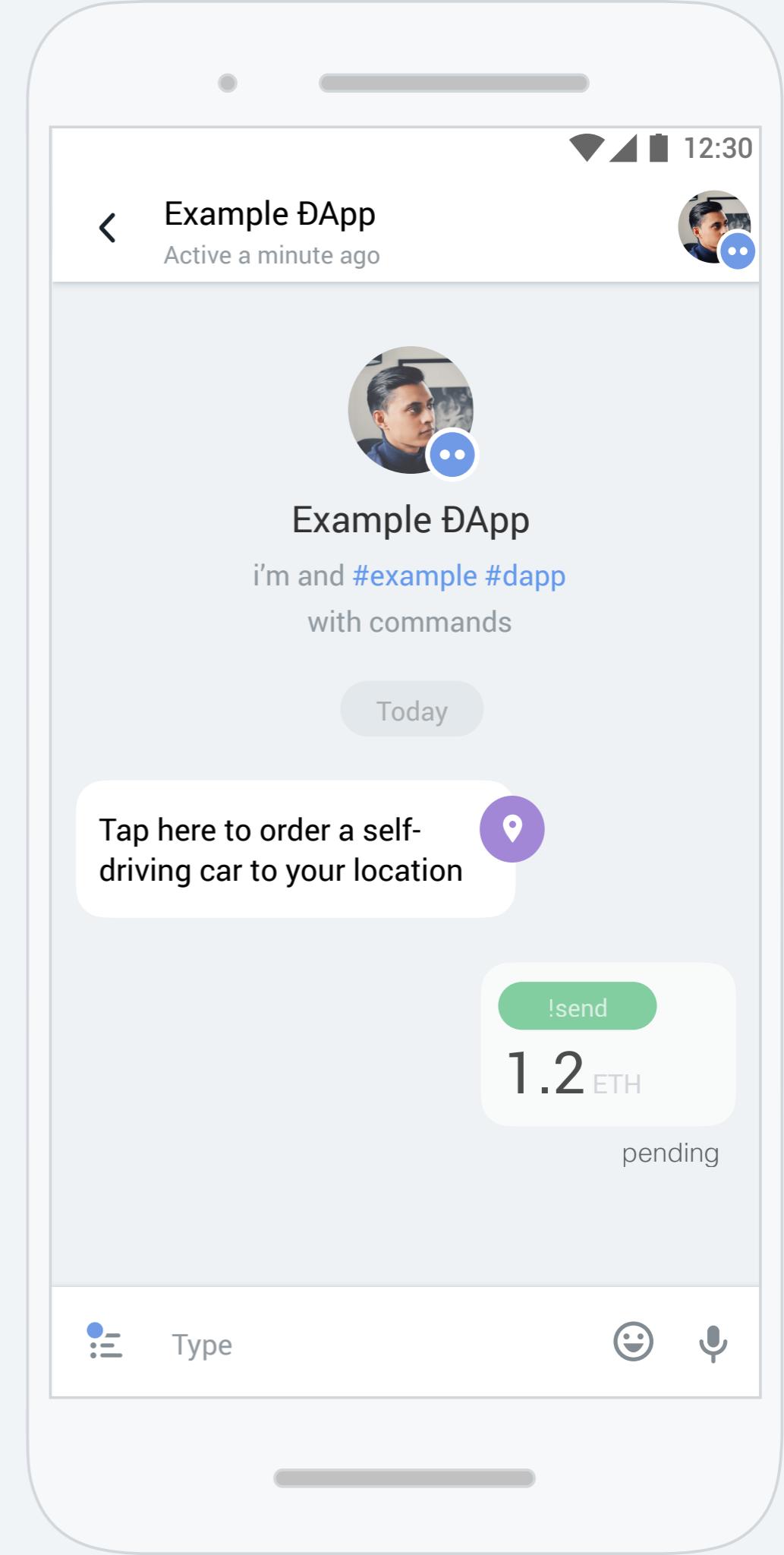
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

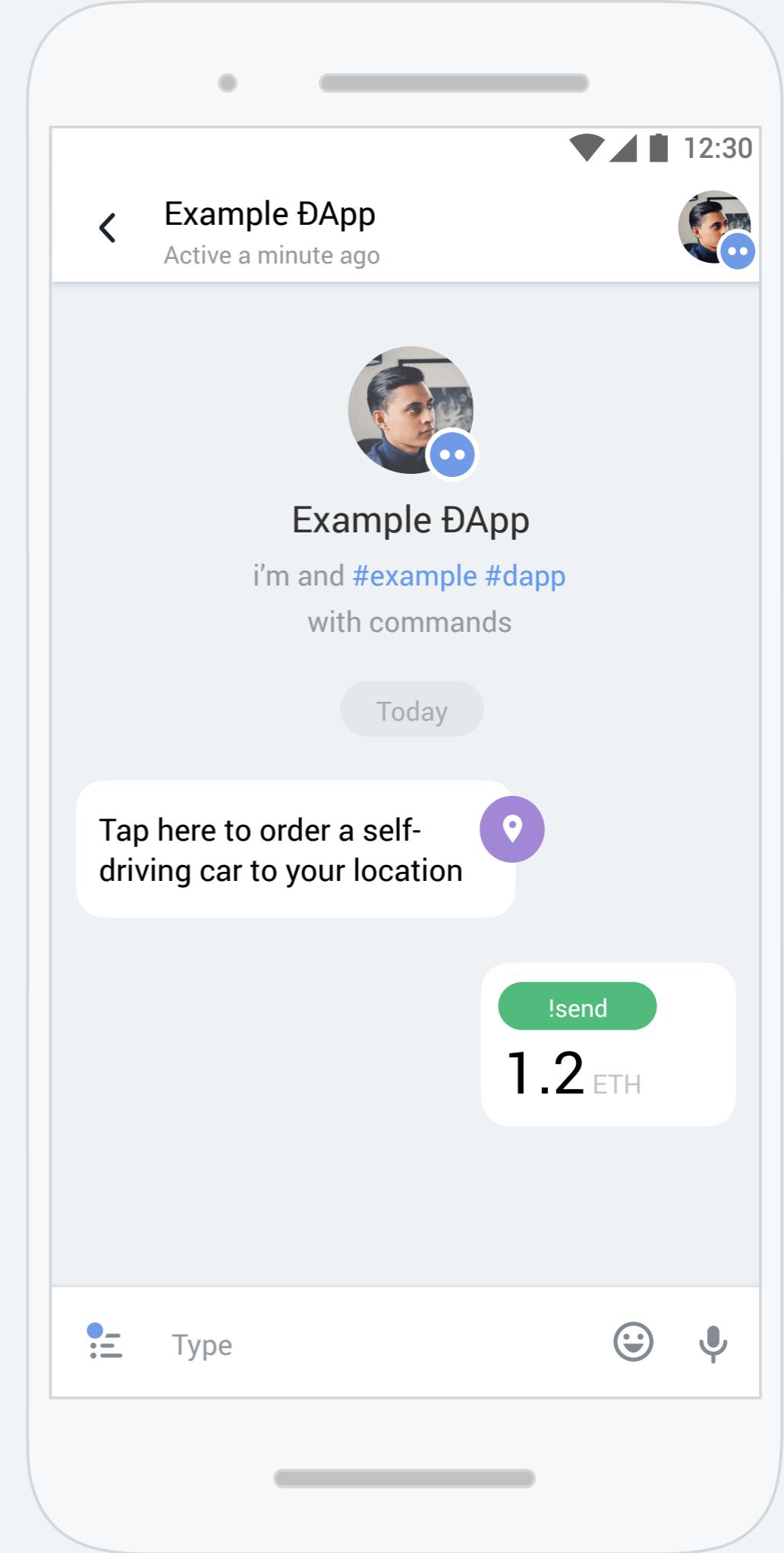
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

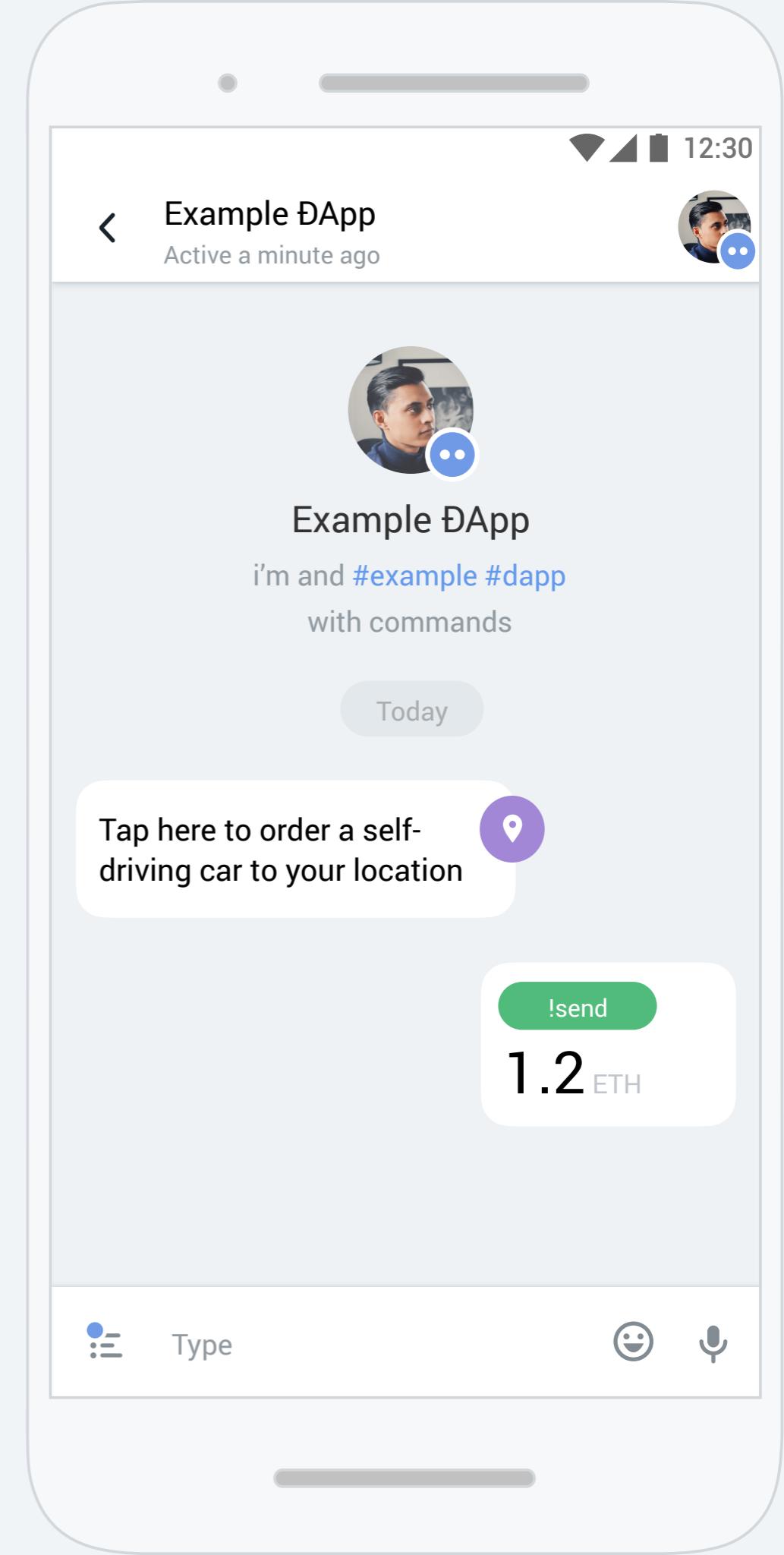
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

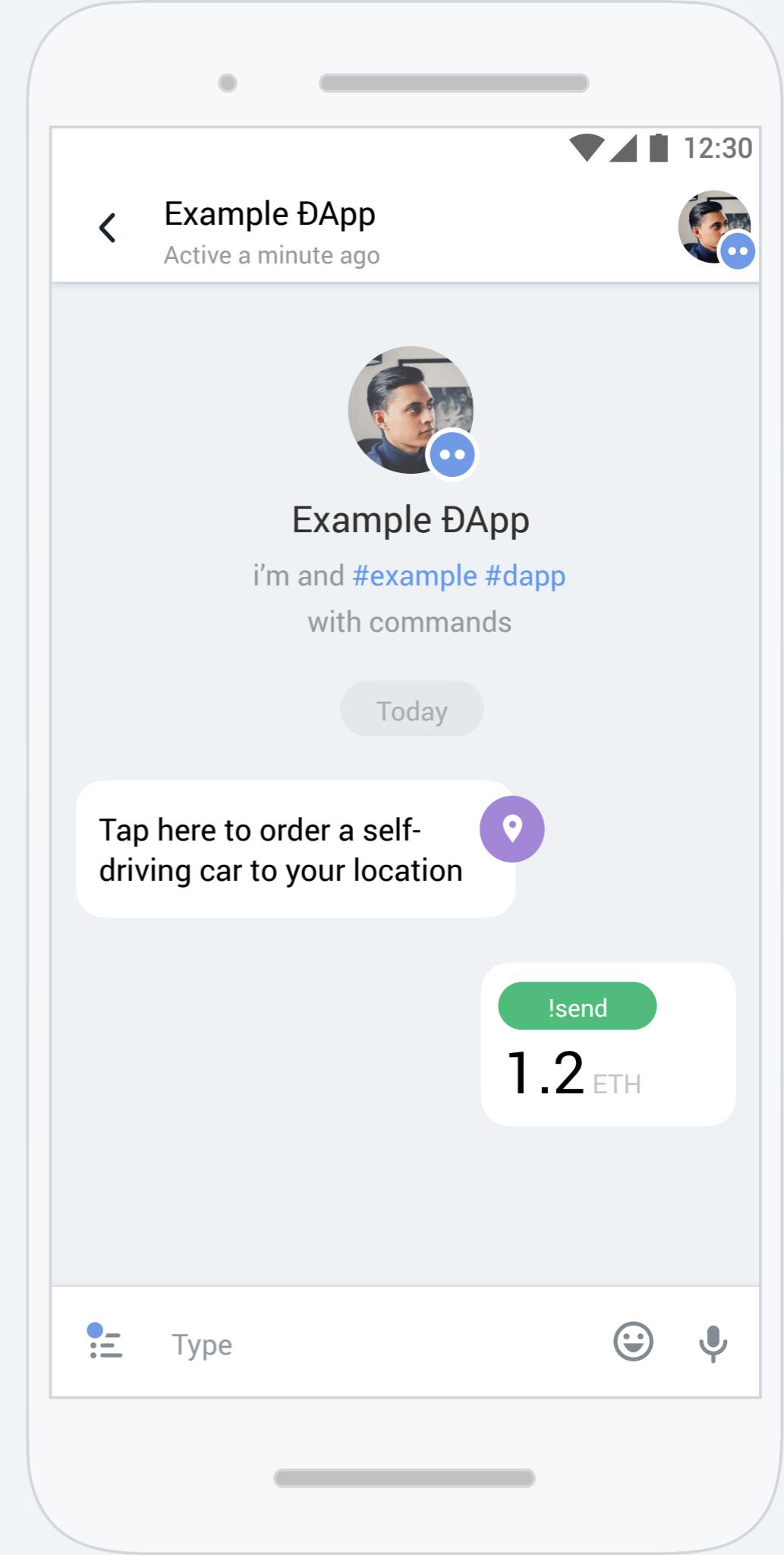
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

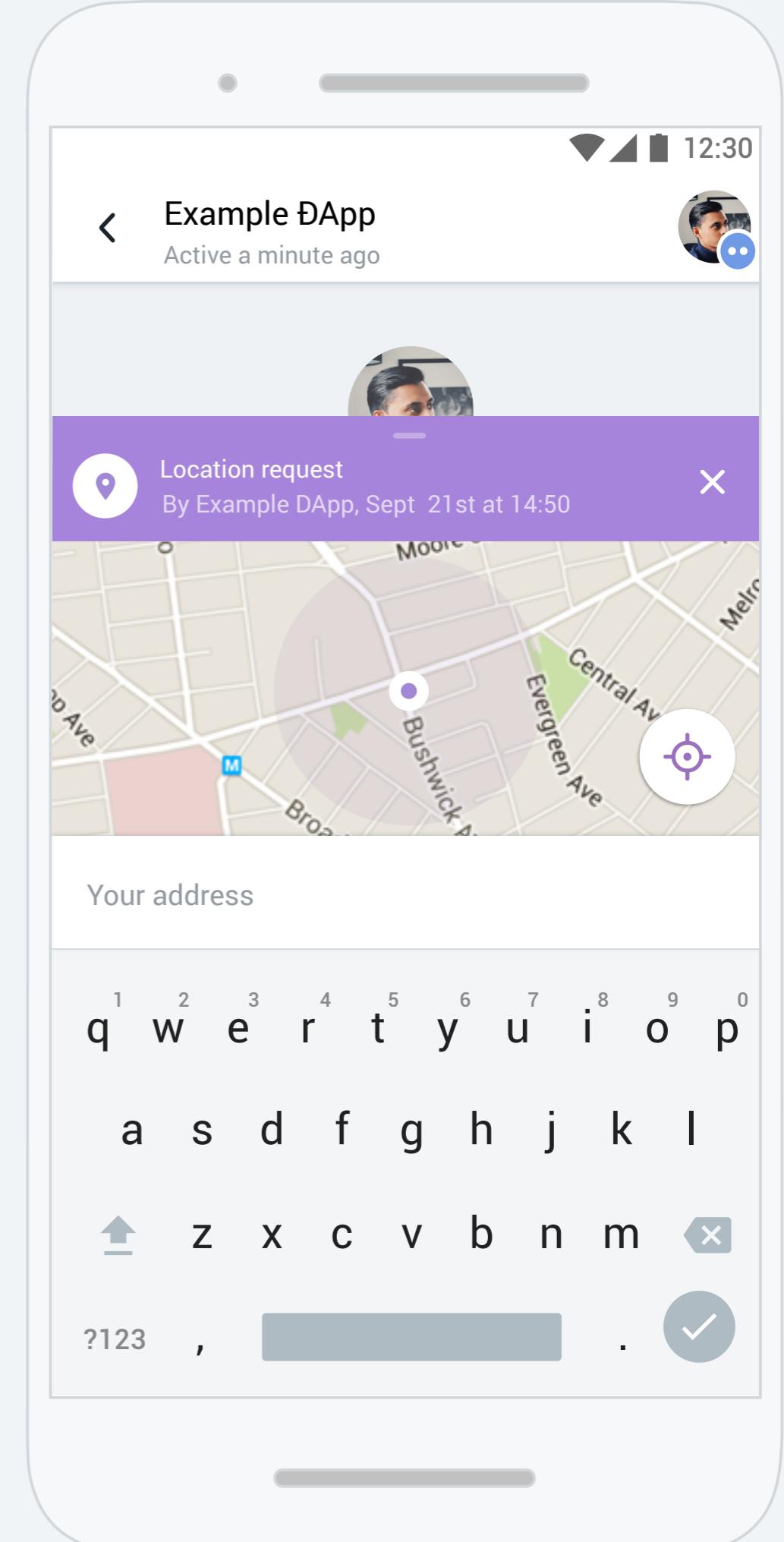
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

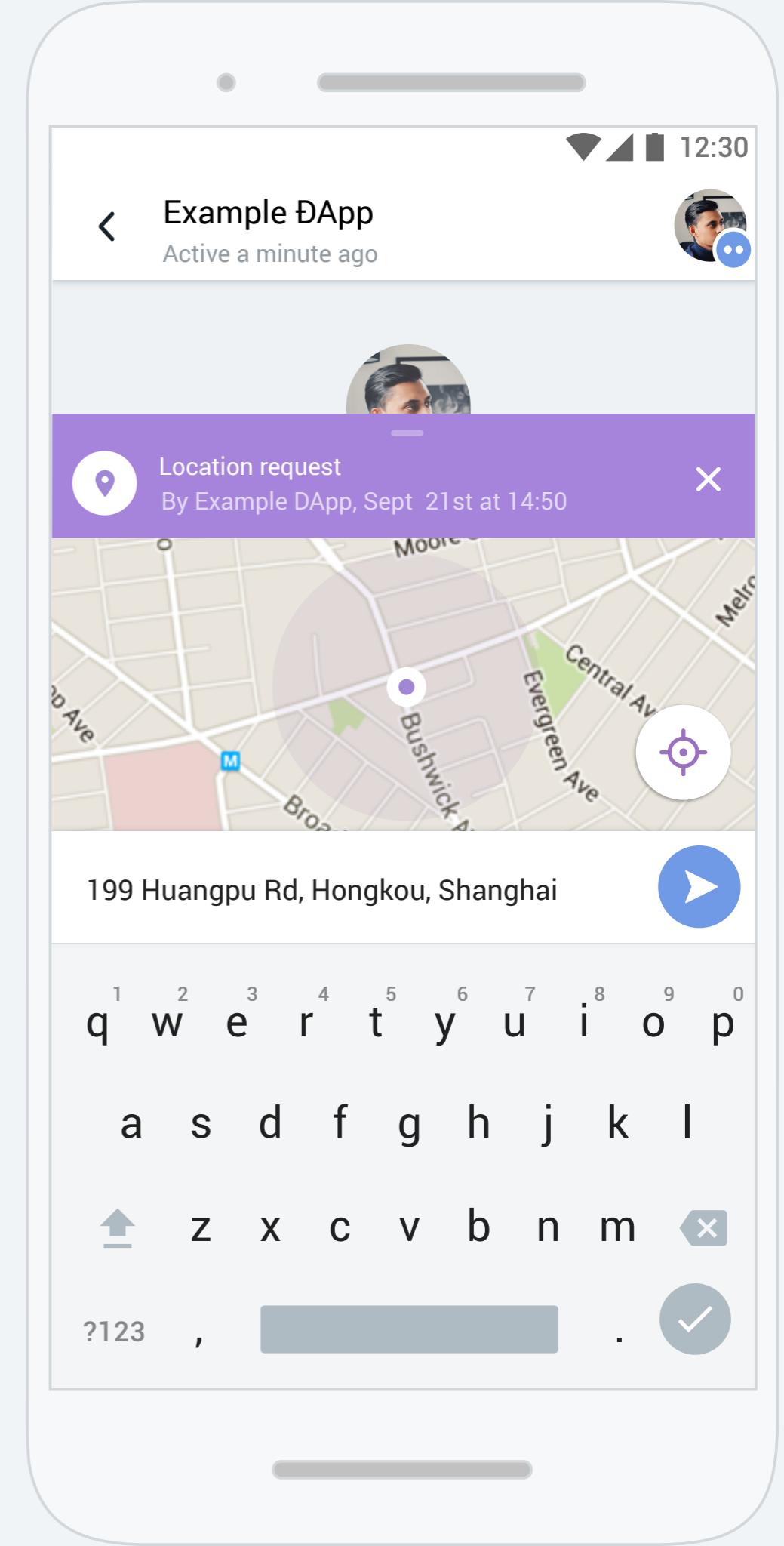
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTACT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

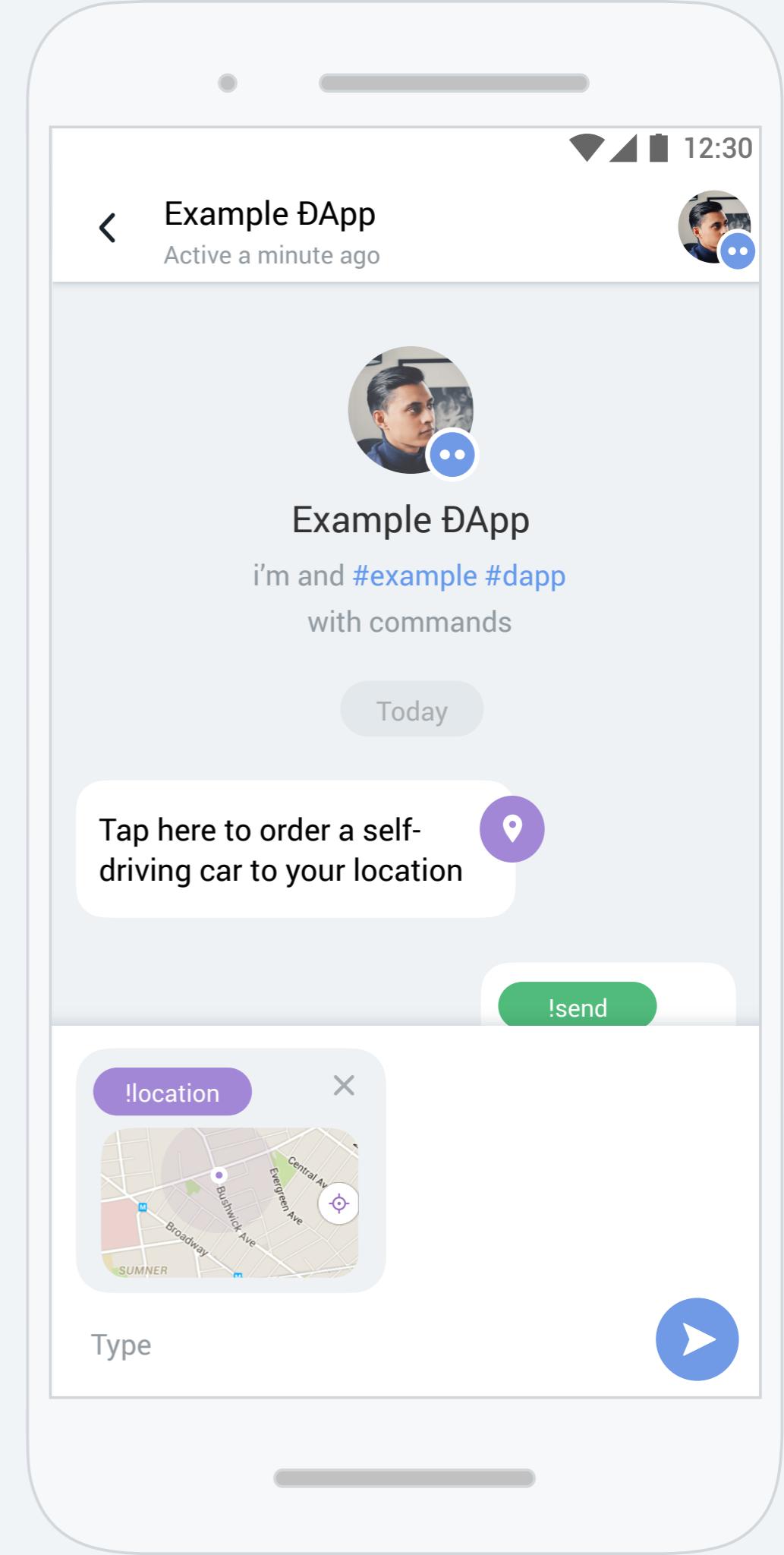
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTACT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

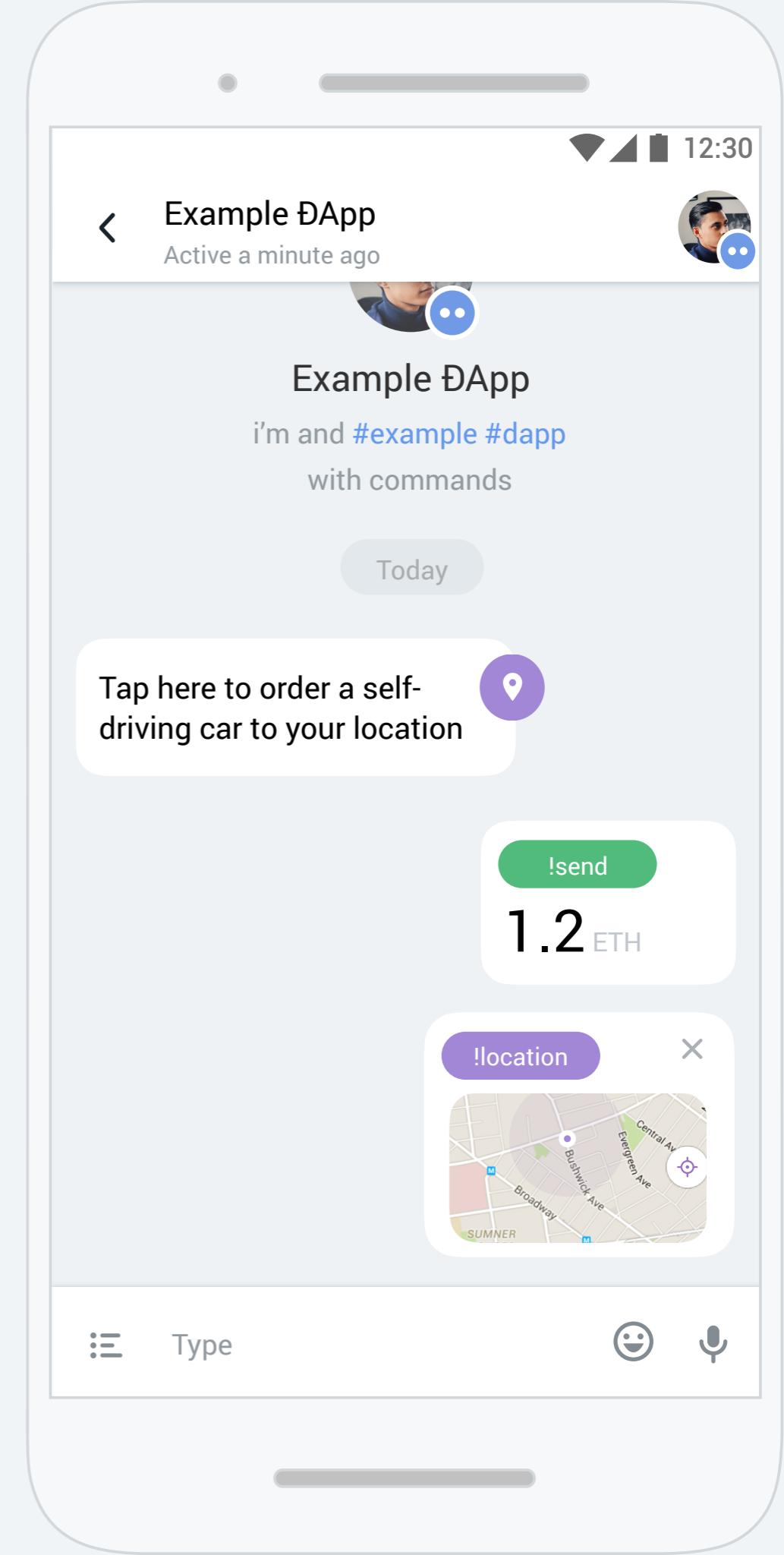
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTEXT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

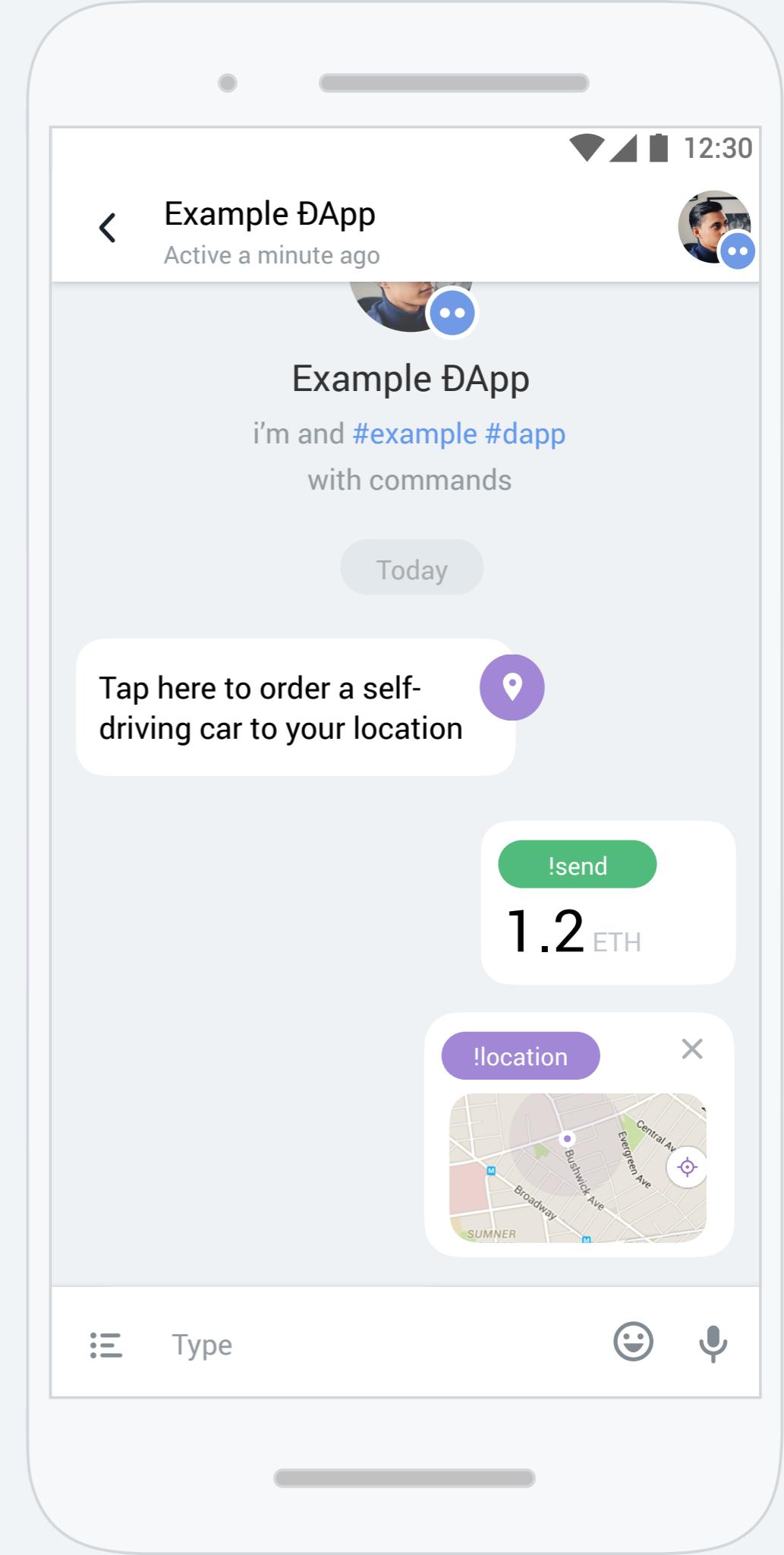
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTACT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

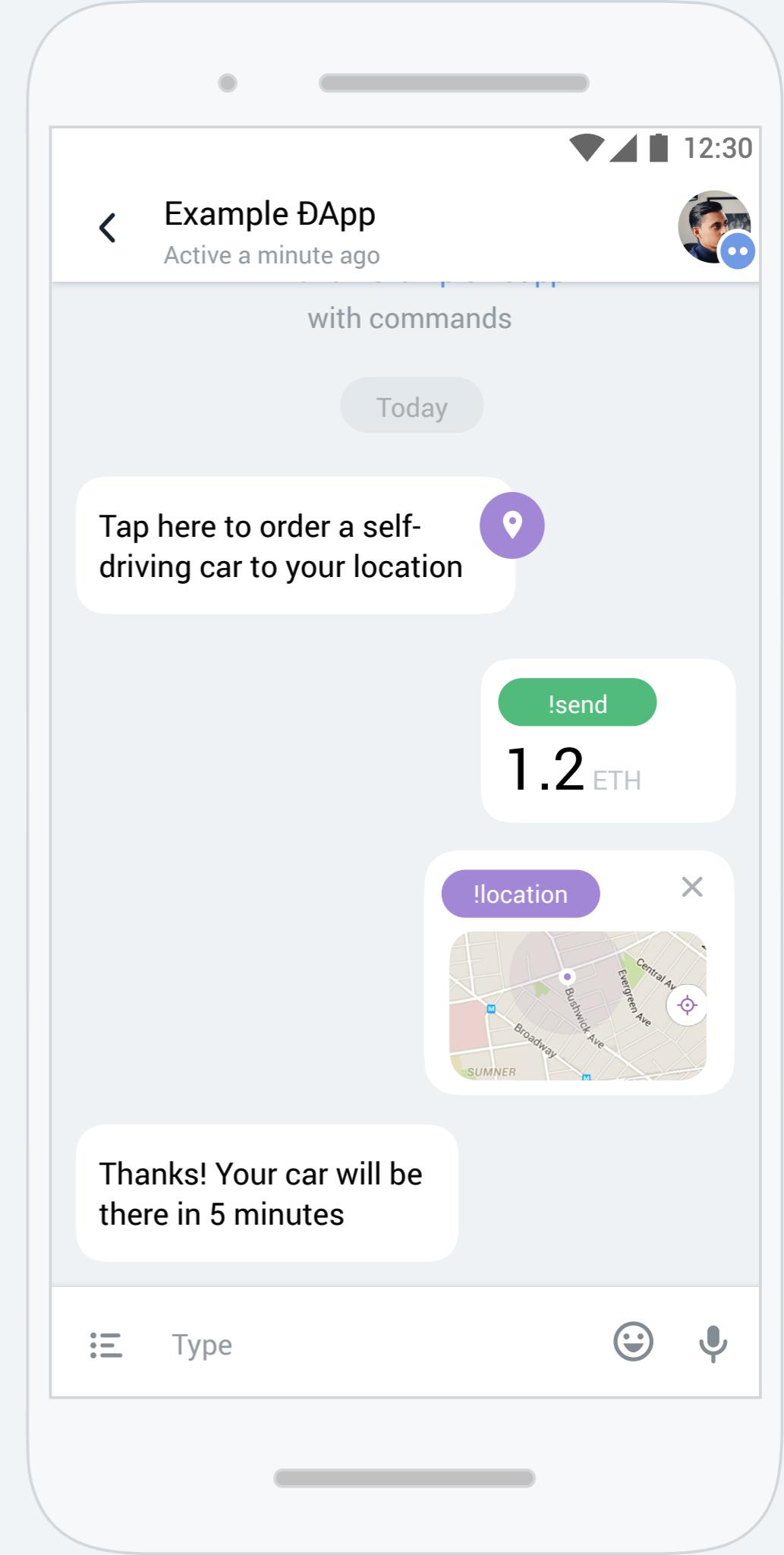
```
var location = {  
    name: "location",  
    color: "#a187d5",  
    description: "Send Location",  
    // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
    name: "whereis",  
    color: "#a187d5",  
    description: "Request a participants Location",  
    params: [  
        {  
            name: "participant",  
            type: status.types.CONTACT,  
            default: context.participants[0]  
        }  
    ],  
    handler: function (params) {  
        status.request({  
            command: 'location',  
            target: params.participant  
        });  
    }  
});
```



# Requests/Responses

daisy-chain commands  
prefill parameters for friends  
for more meaningful conversations

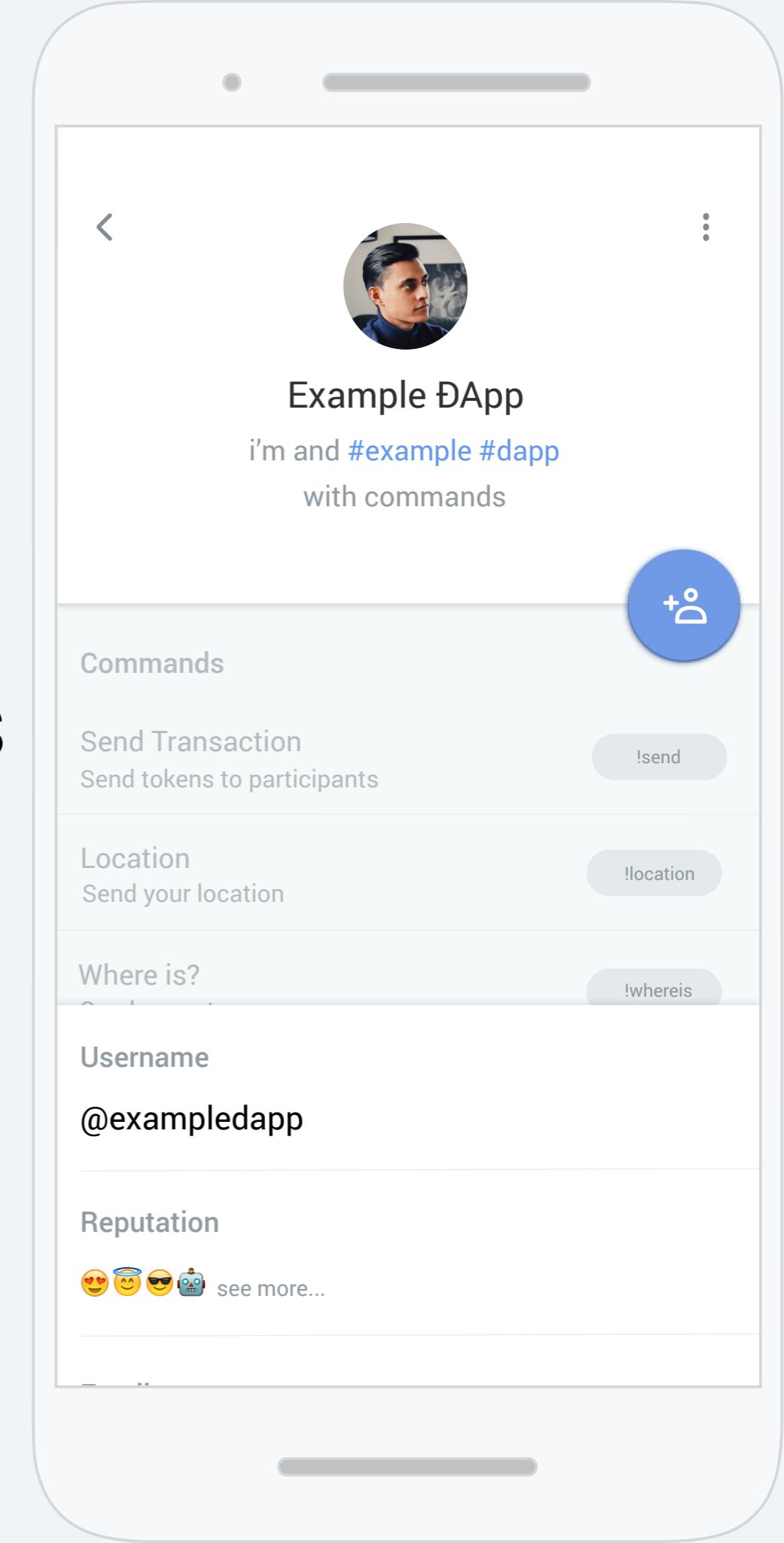
```
var location = {  
  name: "location",  
  color: "#a187d5",  
  description: "Send Location",  
  // ...  
};  
  
status.command(location);  
status.response(location);  
  
status.command({  
  name: "whereis",  
  color: "#a187d5",  
  description: "Request a participants Location",  
  params: [  
    {  
      name: "participant",  
      type: status.types.CONTACT,  
      default: context.participants[0]  
    }  
  ],  
  handler: function (params) {  
    status.request({  
      command: 'location',  
      target: params.participant  
    });  
  }  
});
```





# Command Permissions

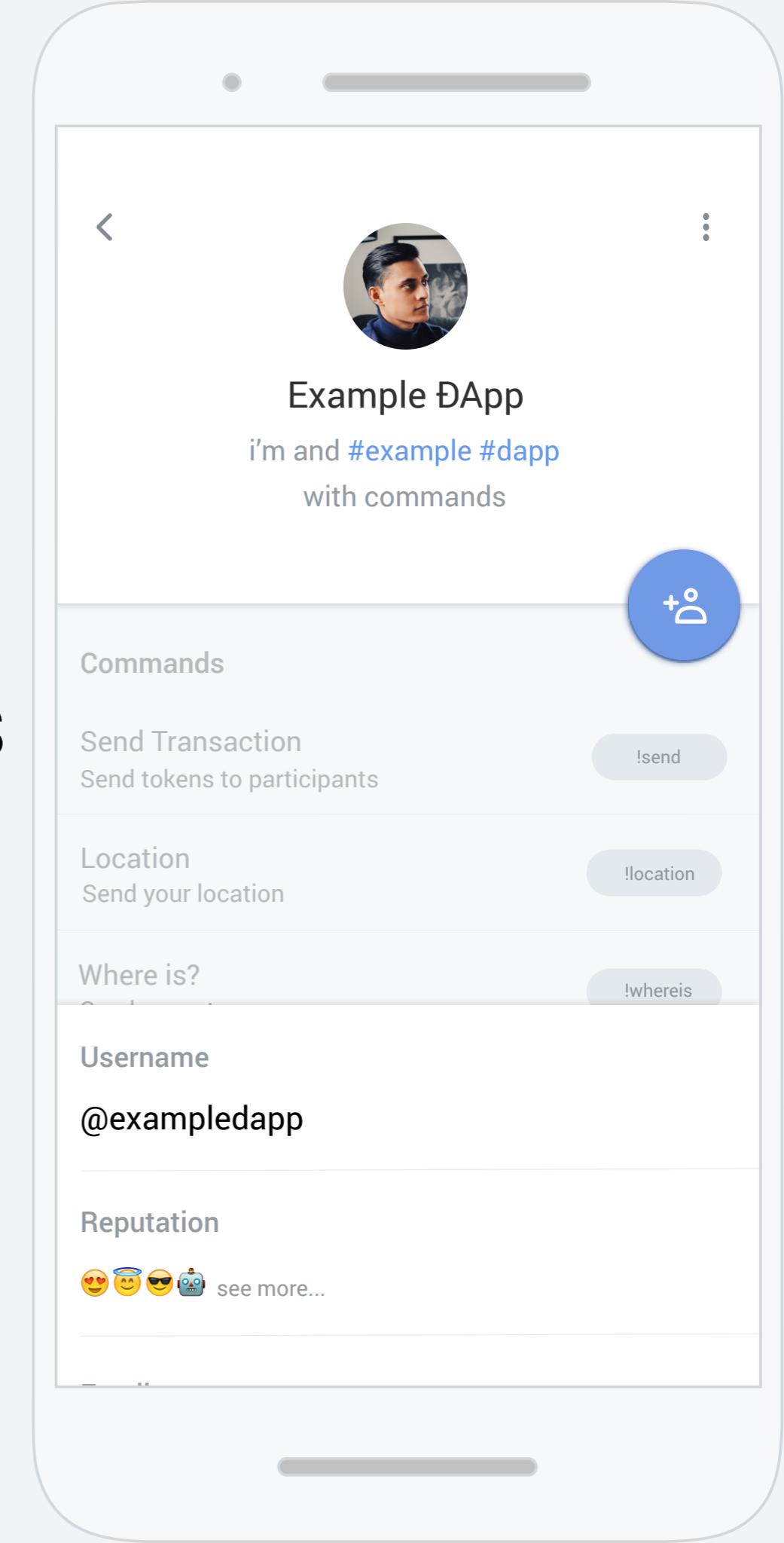
enable commands in group chat





# Command Permissions

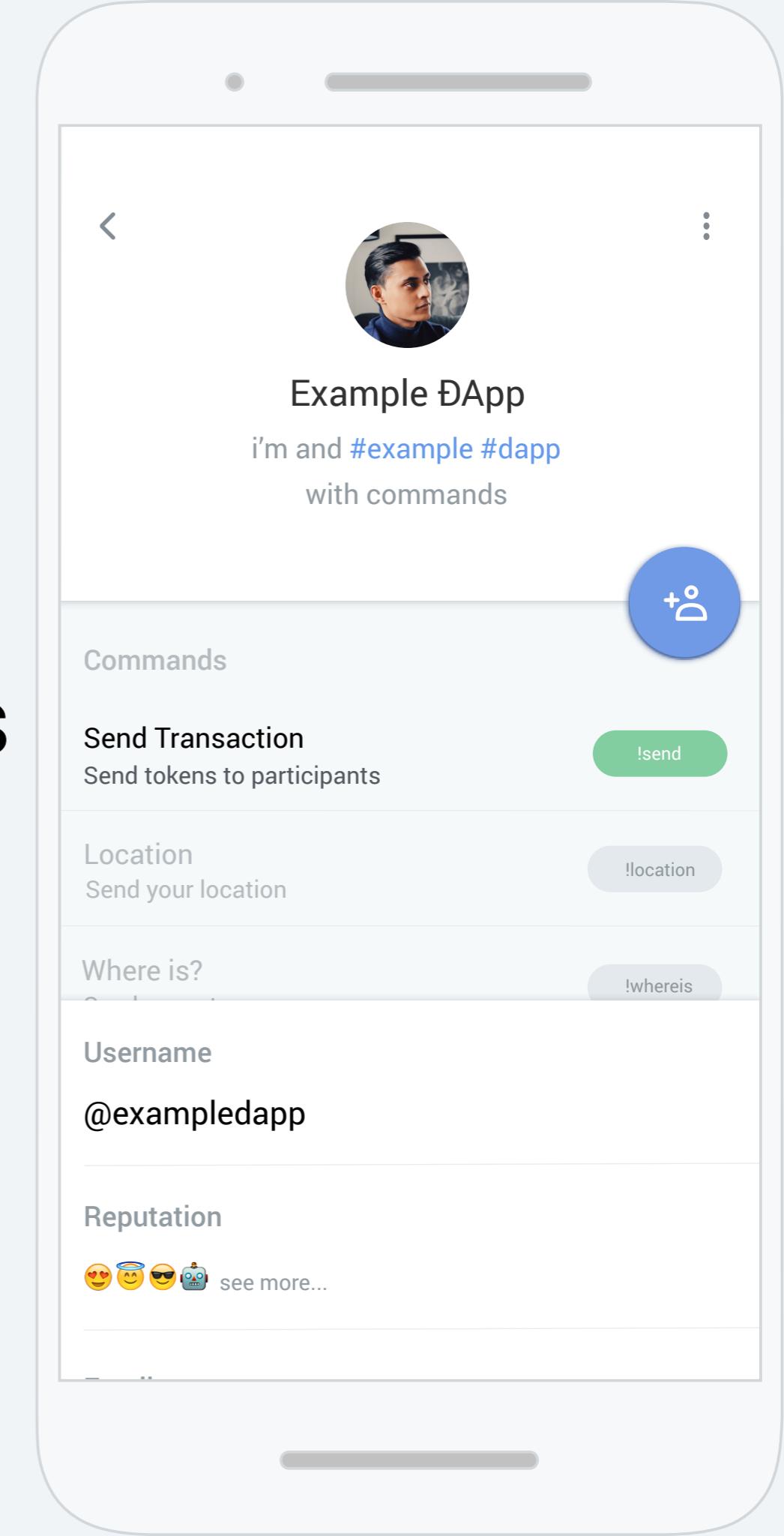
enable commands in group chat





# Command Permissions

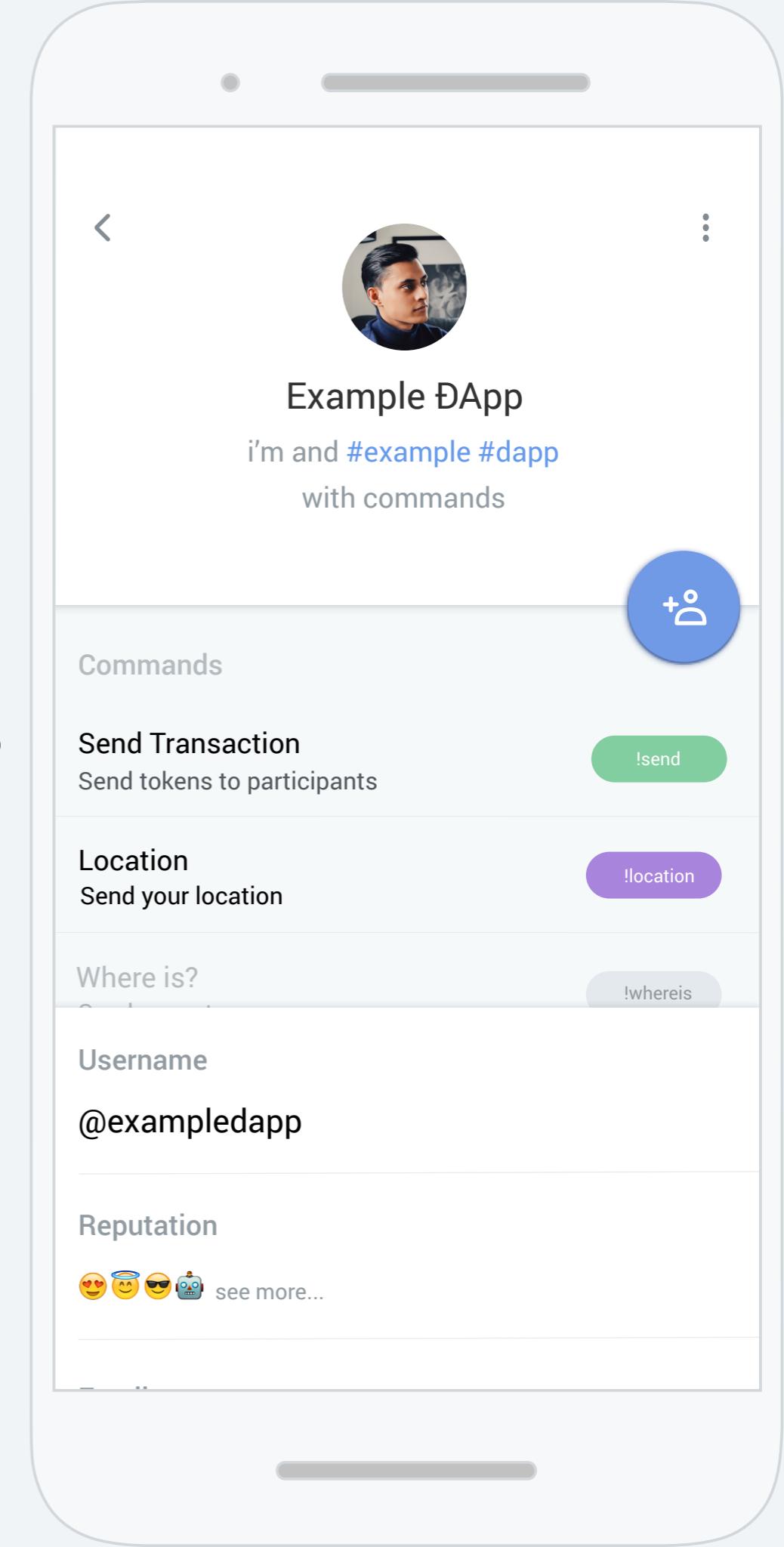
enable commands in group chat





# Command Permissions

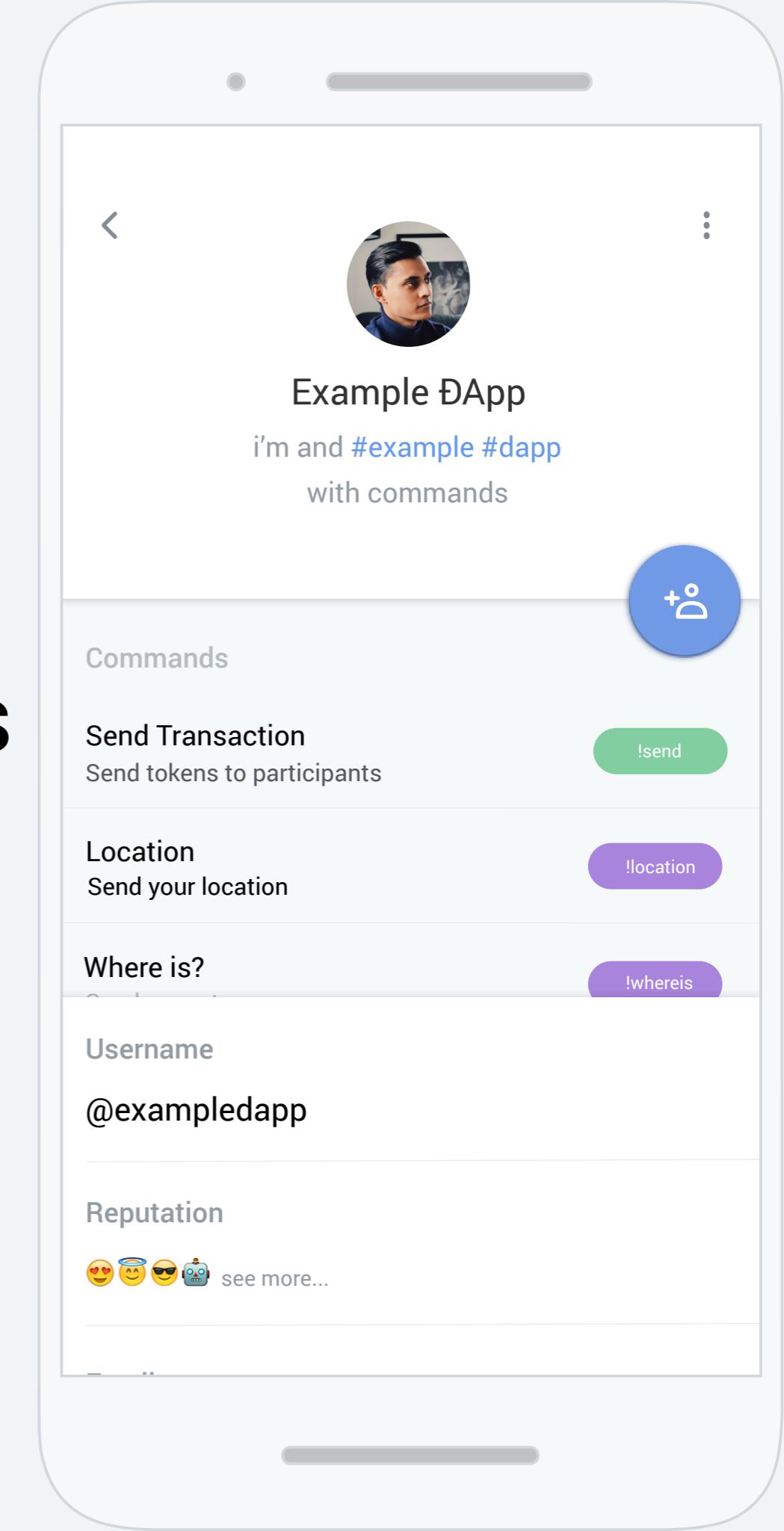
enable commands in group chat





# Command Permissions

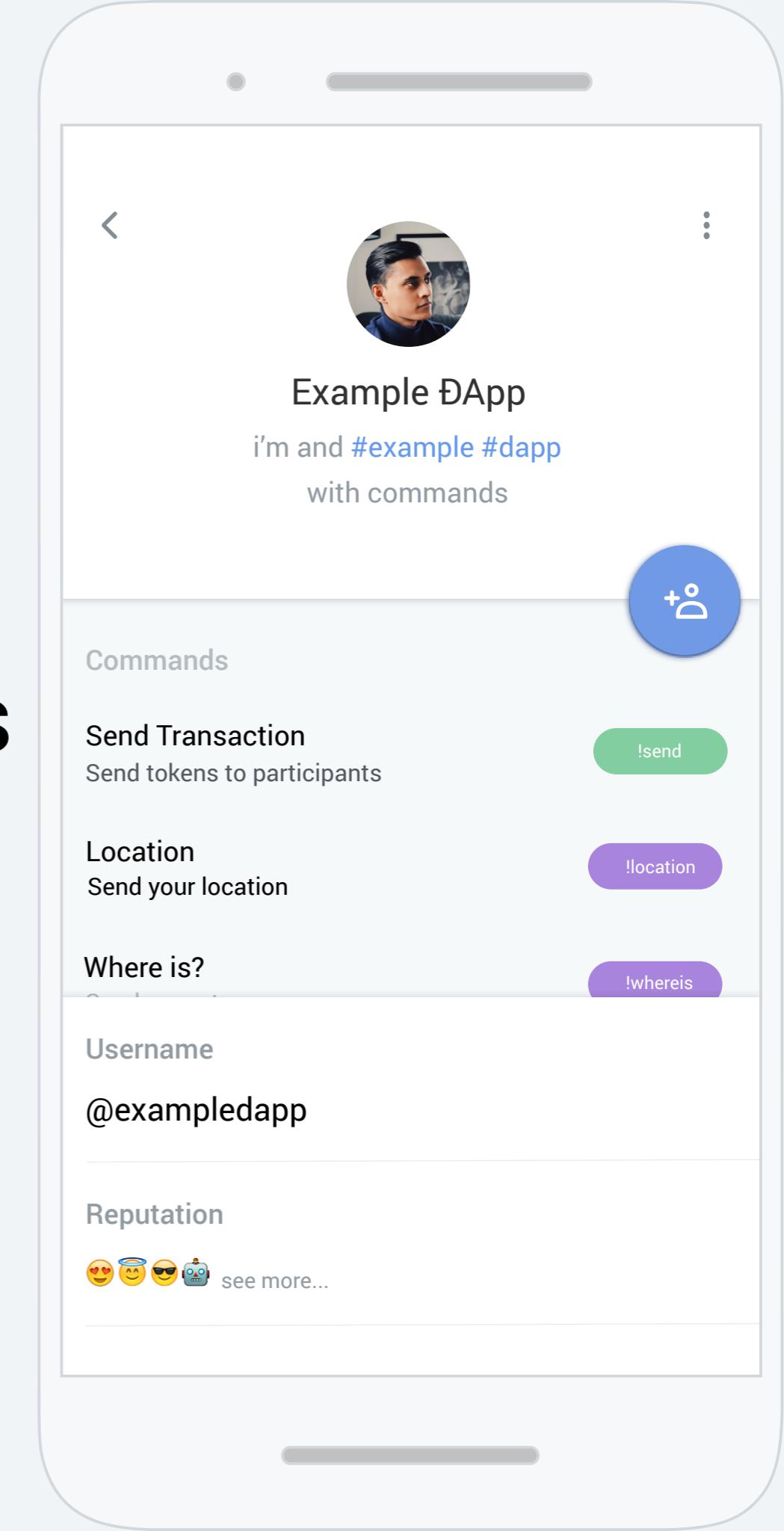
enable commands in group chat





# Command Permissions

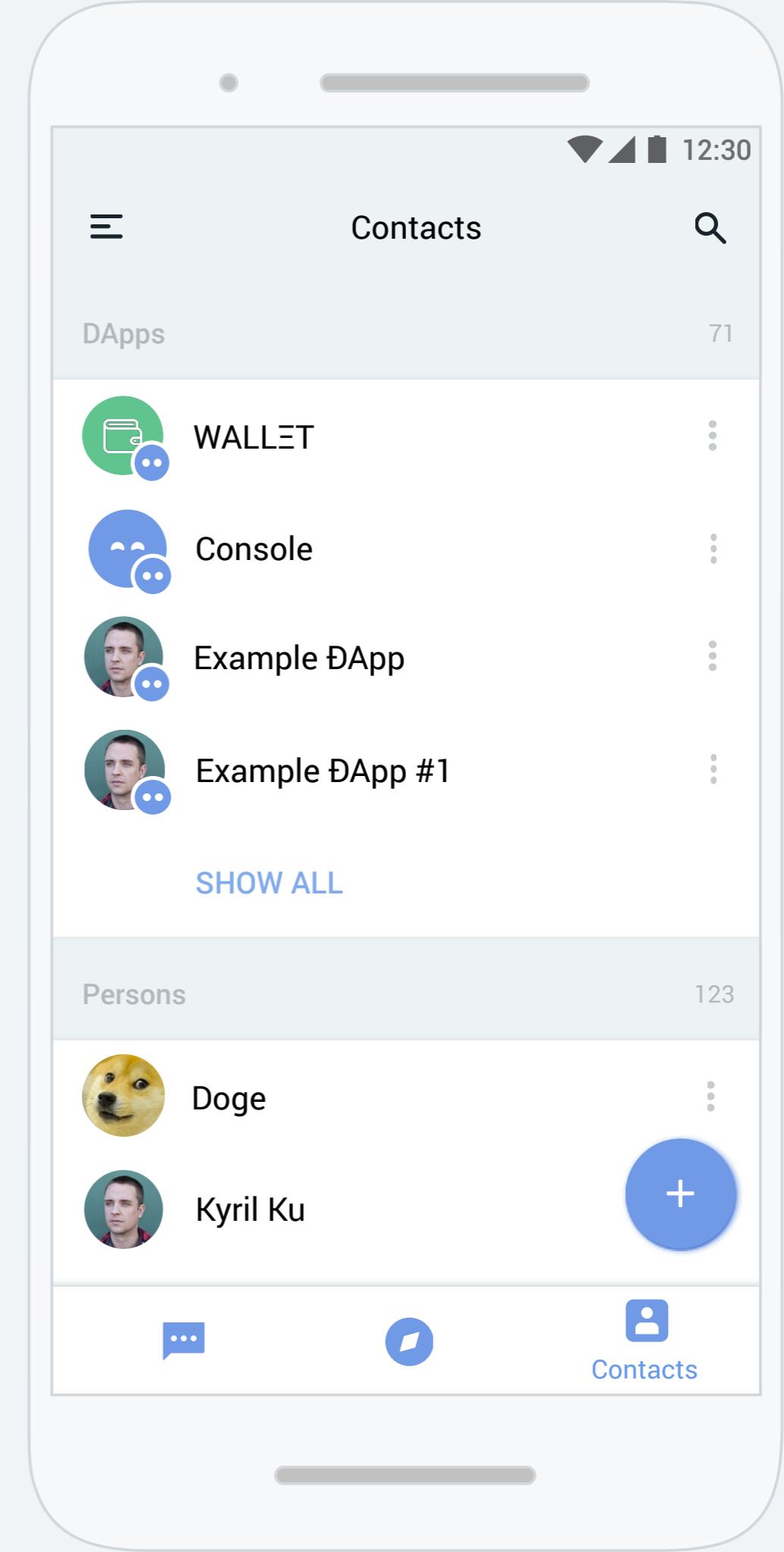
enable commands in group chat



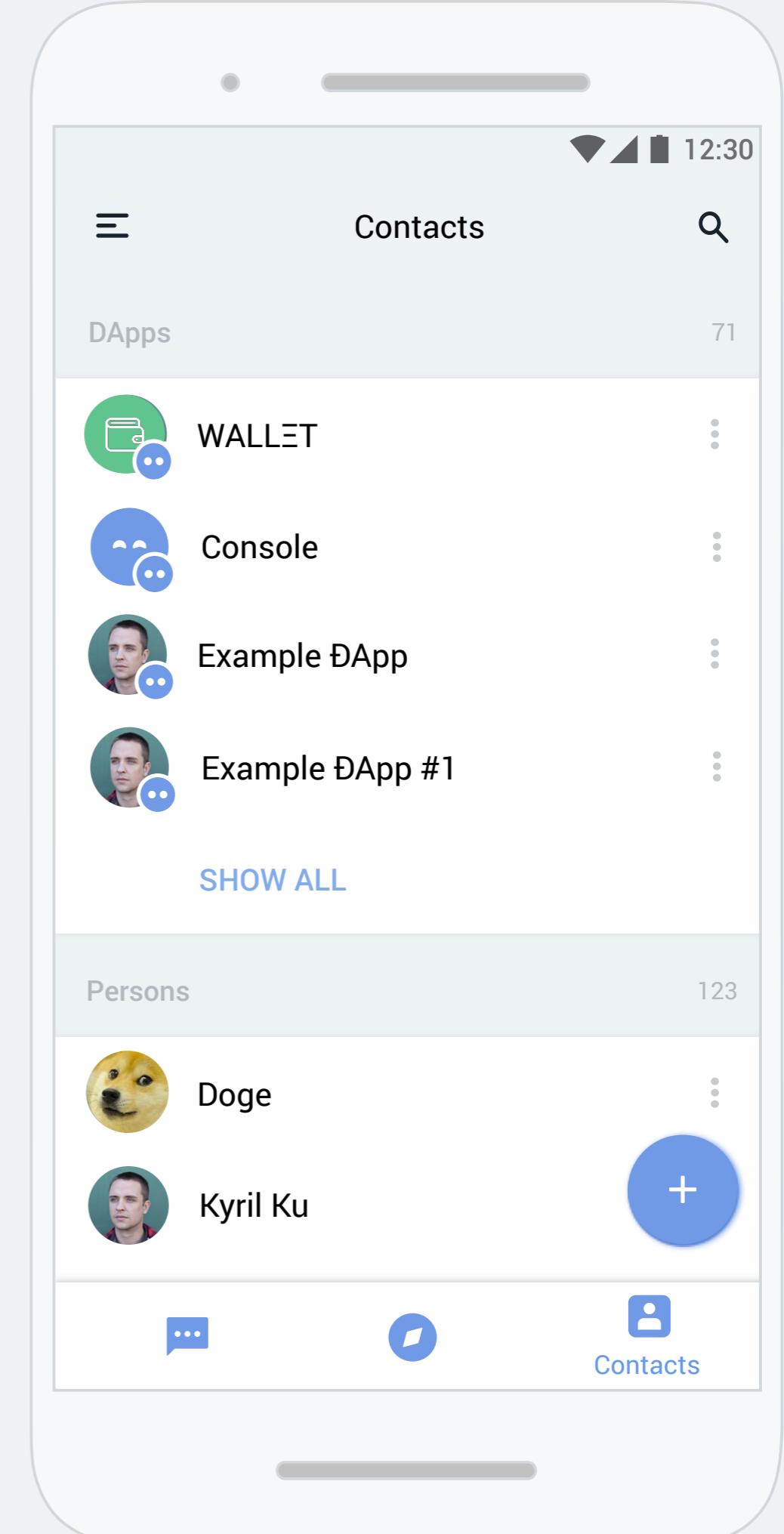


# Command Permissions

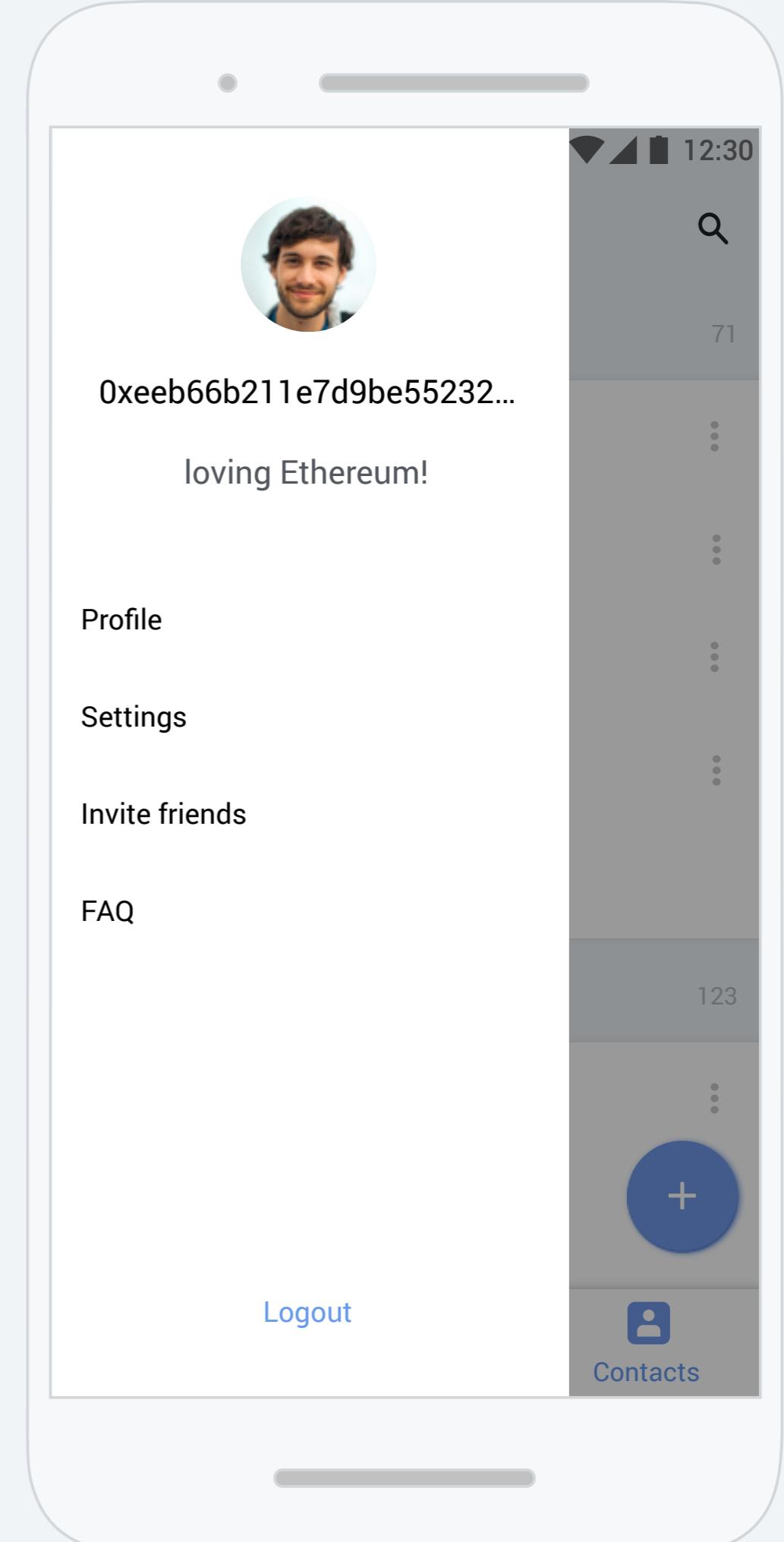
enable commands in group chat



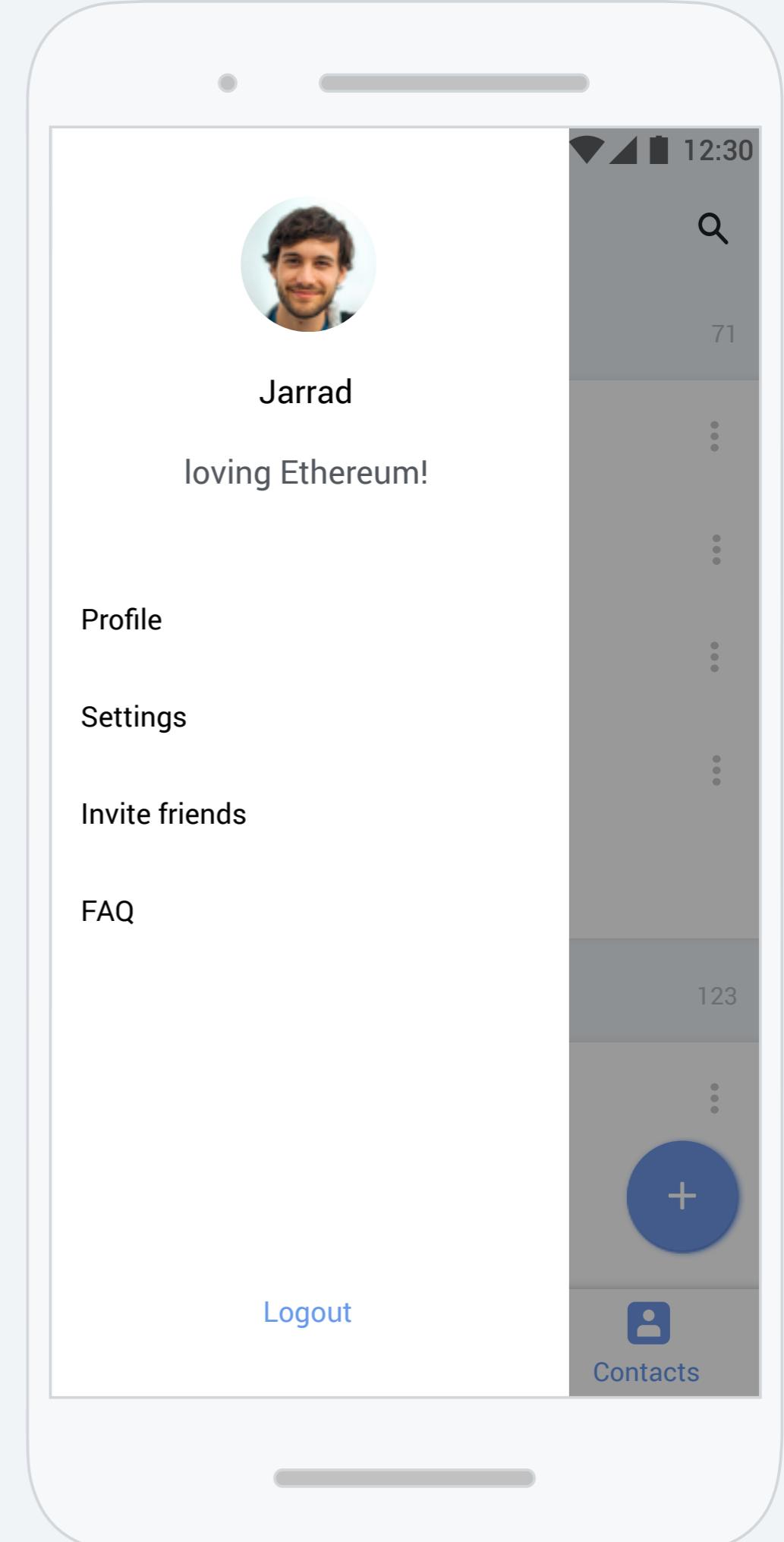
Discover   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



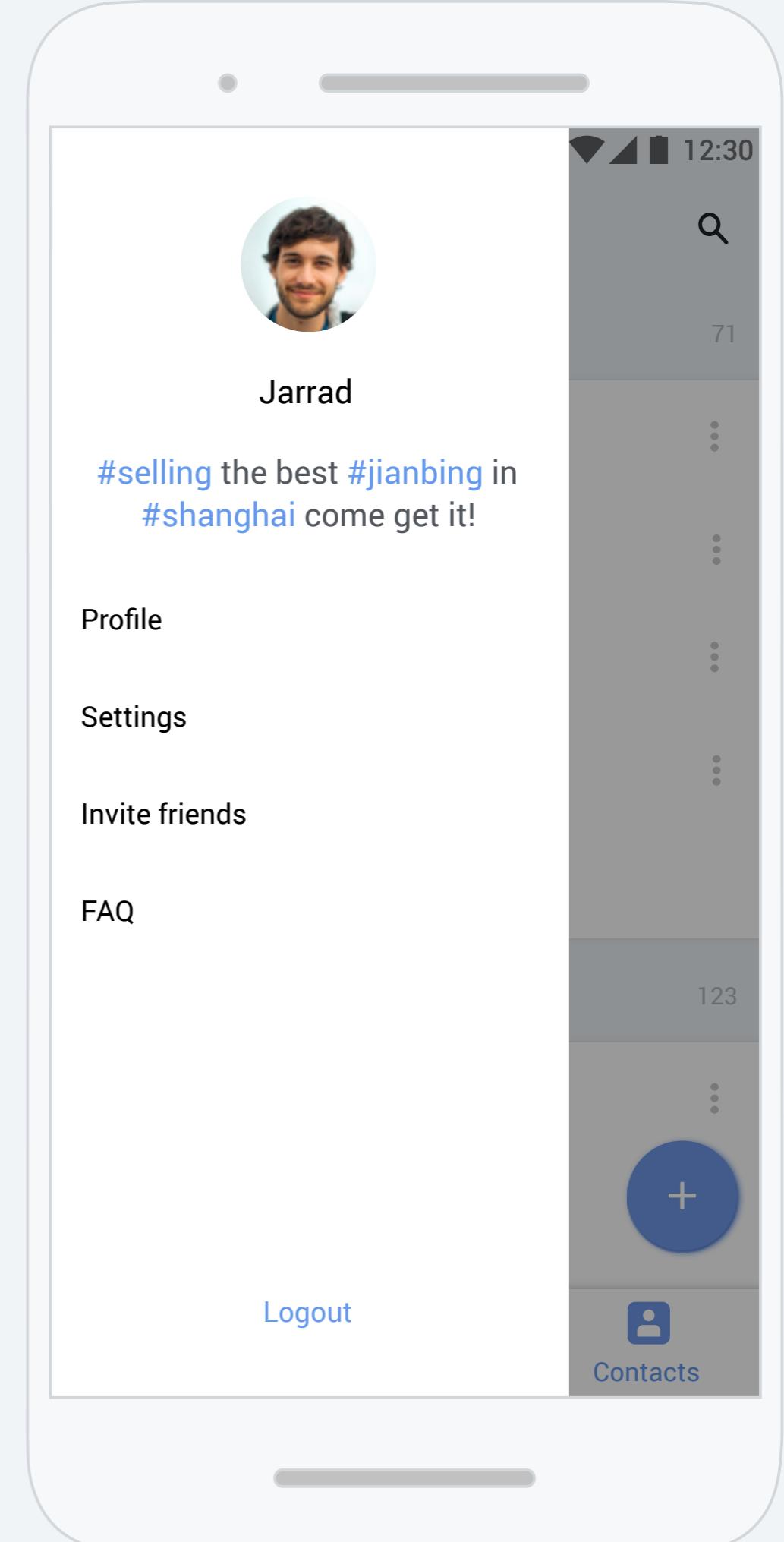
**Discover**   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



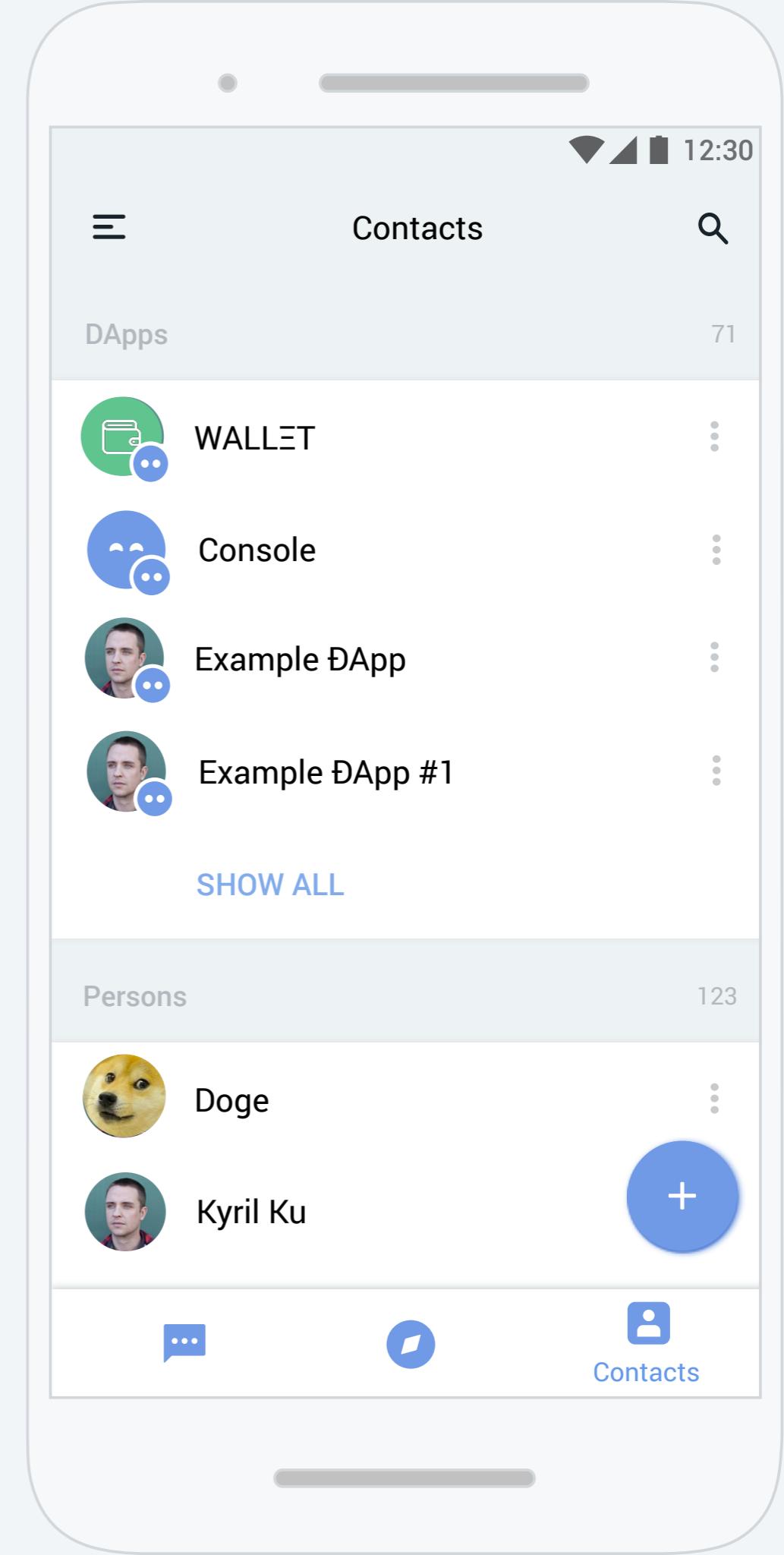
**Discover**   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



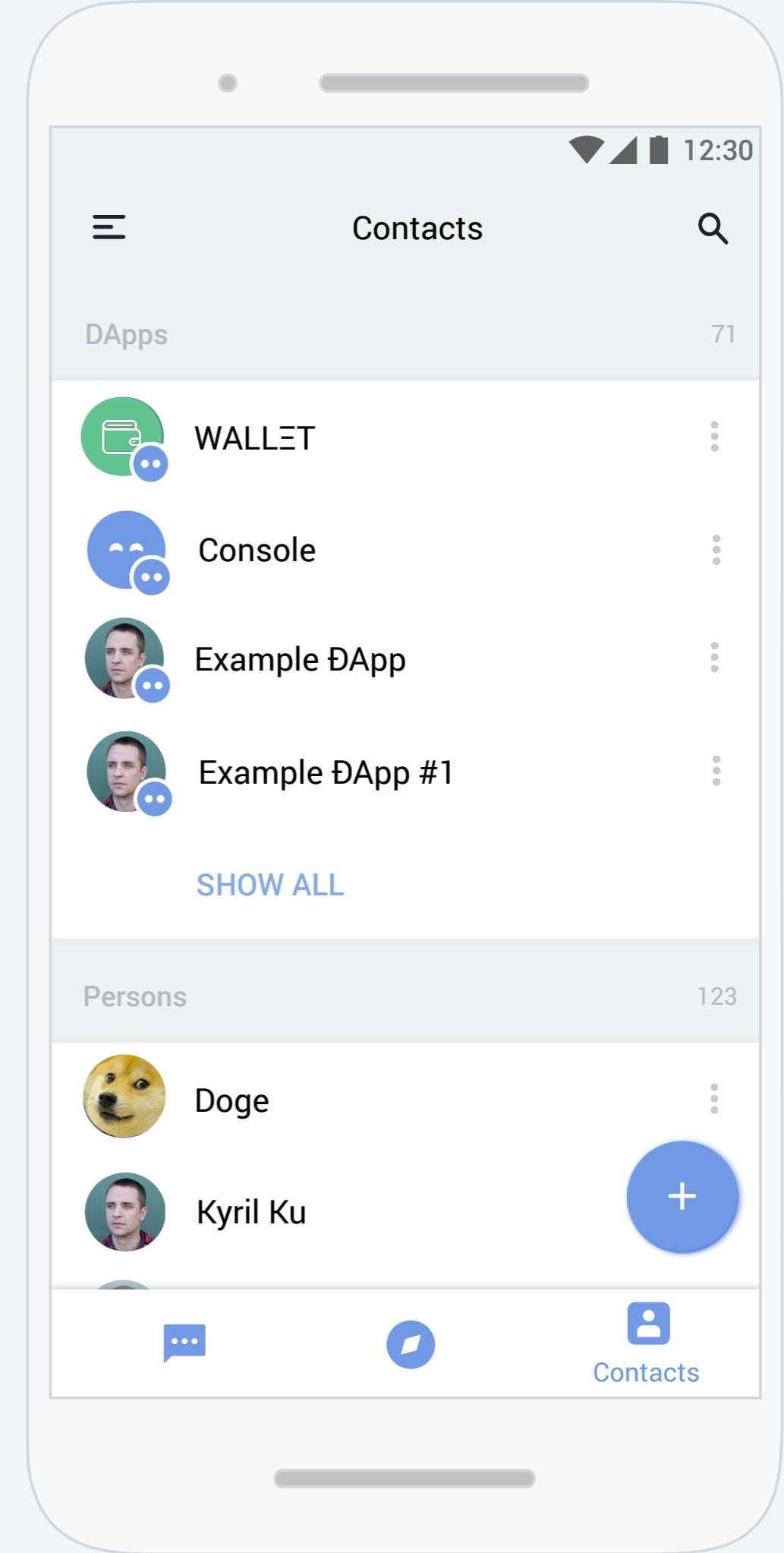
Discover   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



Discover   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



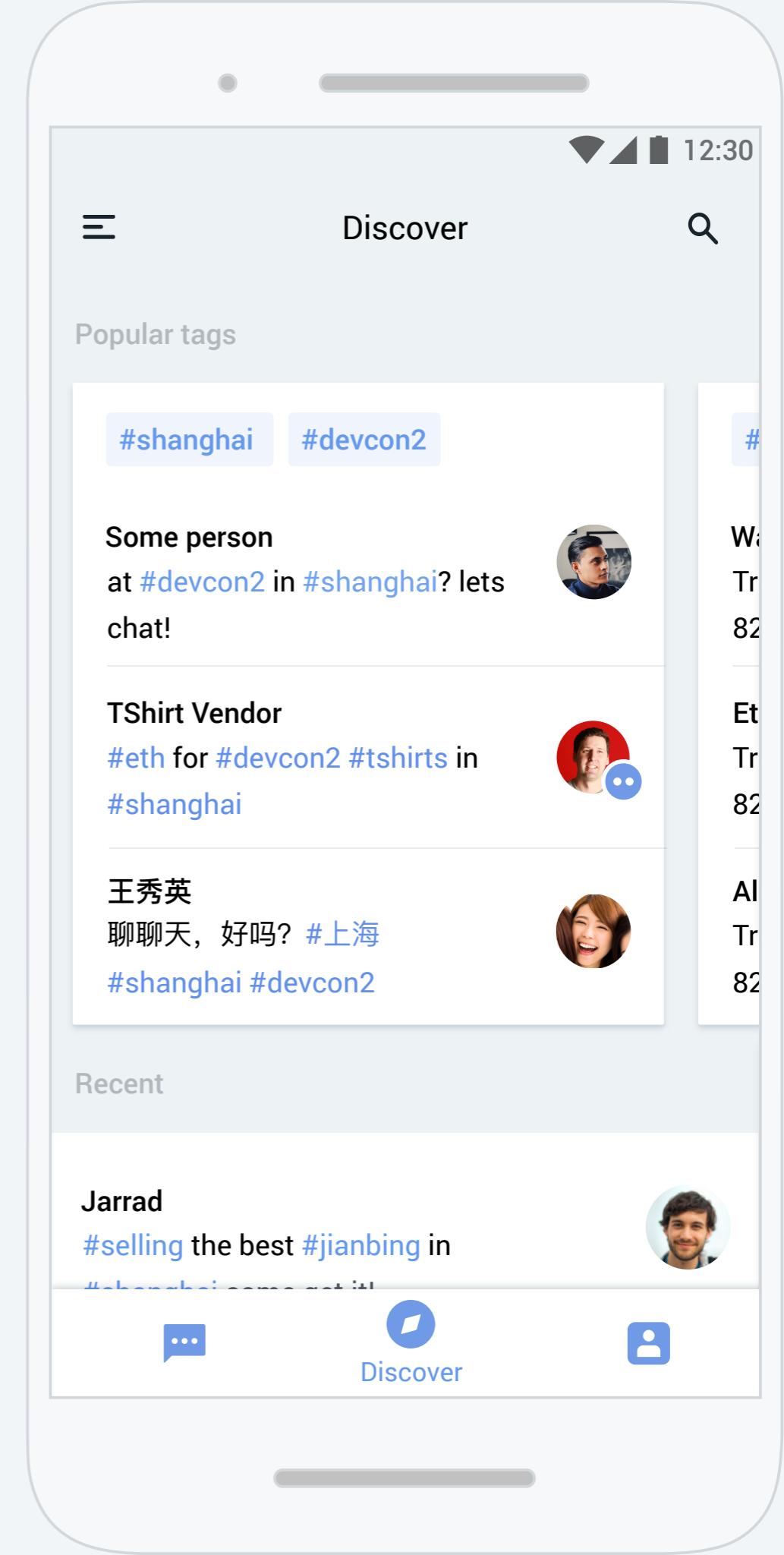
Discover   
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



# Discover



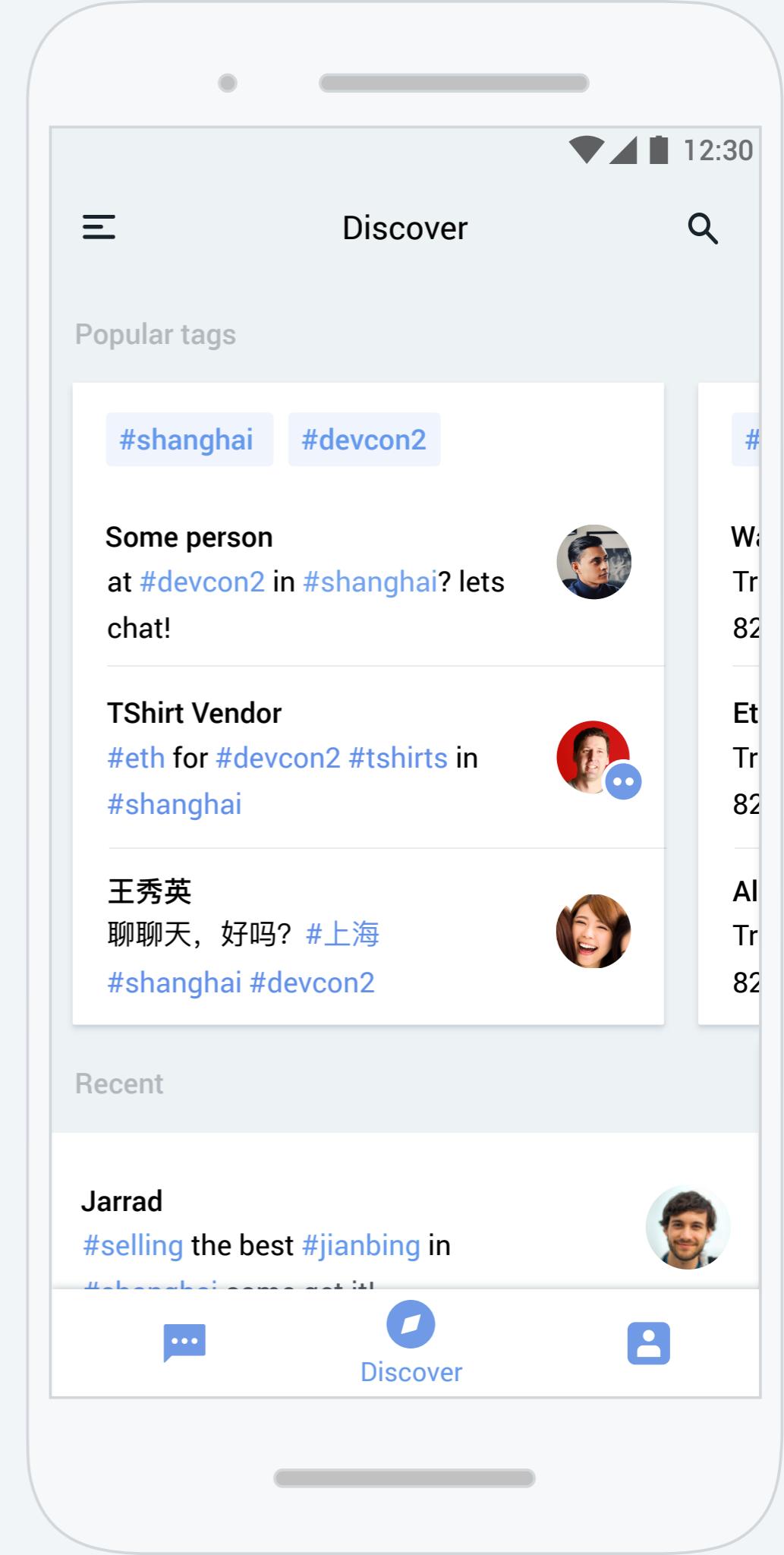
advertise yourself,  
find DApps and participants by  
[#hashtags](#)



# Discover



advertise yourself,  
find DApps and participants by  
[#hashtags](#)



# Chat API Extensions



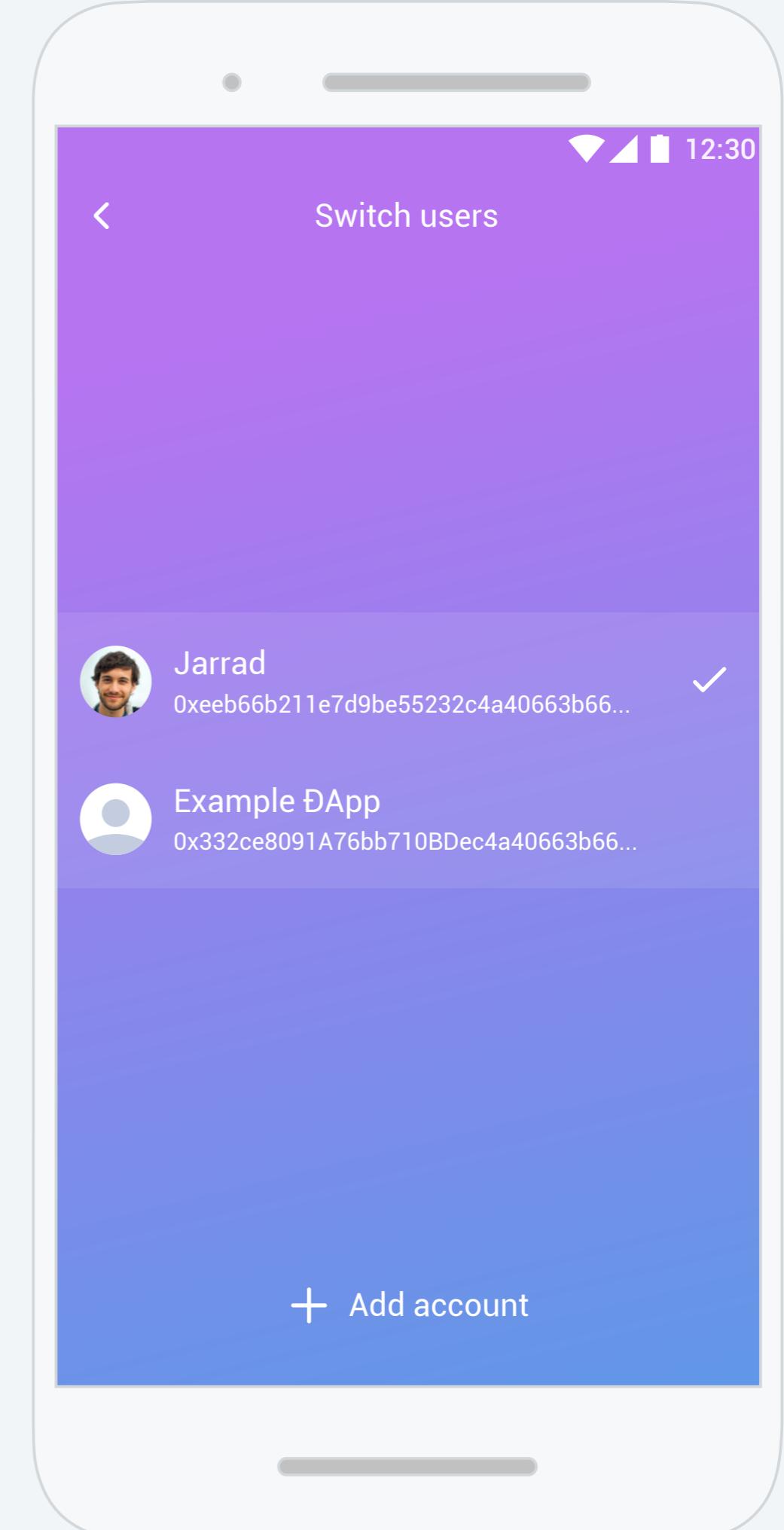
QR Scanning / Generation  
Contact Menu Creation



# Multi-account ready



for shared devices  
and DApp developers  
or just to logout of app





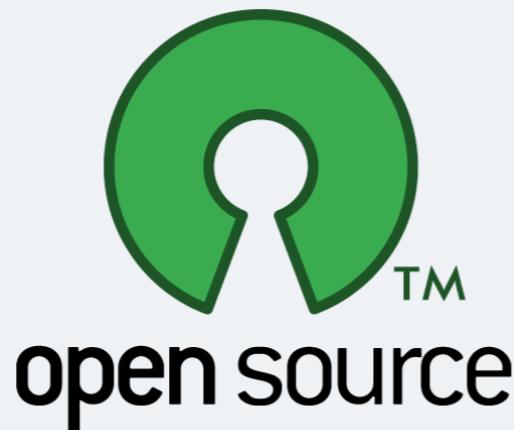
## current state of status

base system 'feature complete'

moving into testing phase

binary by end of year.. 🙏

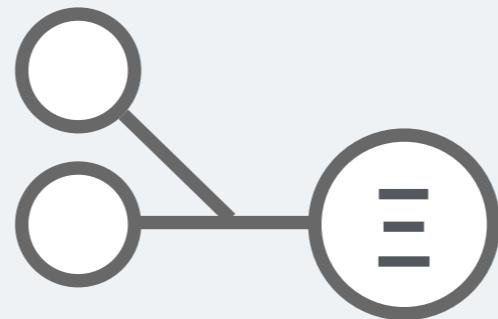
# Build it yourself! 🏋️‍♂️ 😅



clojurescript	
react native	
objective-c	java
status-go	
go-ethereum	



<https://github.com/status-im/status-react/wiki/Developers'-Guide>



# commiteth

The screenshot shows a GitHub issue page for a repository named 'status-im / status-react'. The issue is titled 'Fix bug in XYZ #1'. A comment from 'jarradh' states: 'There is a bug in XYZ, can someone please fix it! Steps to reproduce; • open application • tap on settings • red screen'. Another comment from 'commiteth' includes a QR code and the text: 'commiteth 45.00 ETH 0x89205A3A3b2A69De6Dbf7f01ED13B2108B2c43e7 status-im/status-react/issues/101'.

Fix bug in XYZ #1

Open jarradh opened this issue 2 days ago · 1 comment

jarradh commented 2 days ago • edited

Owner +

There is a bug in XYZ, can someone please fix it!

Steps to reproduce;

- open application
- tap on settings
- red screen

jarradh added labels 11 minutes ago

commiteth commented 11 minutes ago

+

commiteth  
45.00 ETH  
0x89205A3A3b2A69De6Dbf7f01ED13B2108B2c43e7  
status-im/status-react/issues/101

Labels

bounty  
bug

Milestone

No milestone

Assignees

No one—assign yourself

2 participants

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.



thank-you

<https://status.im>

<http://slack.status.im/>

twitter: @ethstatus @jarradhope

[jarrad@status.im](mailto:jarrad@status.im)

