

@jarolrod coredev berlin

**./bitcoin-qt**

# 2009

Bitcoin.org

Bitcoin v0.1 Alpha Release

Announcing the first release of Bitcoin, a new open source peer-to-peer electronic cash system that's completely decentralized, with no central server or trusted parties. Users hold the crypto keys to their own money and transact directly with each other, with the help of the network to check for double-spending.

Bitcoin

File Options Help

Send Coins Address Book

Your Bitcoin Address: 19CncWdnU57yq4QHBNVPdScFB54JkPGu28 Copy to Clipboard

Balance: 15.05

All Transactions

Status	Date	Description	Debit	Credit
0/unconfirmed 7 blocks	03/01/2009 23:45 03/01/2009 22:18	To: Alice 192.168.0.12 - Order #12345 From: Alice - Thanks for testing all that software	-9.95	+25.00

3 connections 213 blocks 2 transactions

Send Coins

Enter the recipient's IP address (e.g. 123.45.6.7) for online transfer with comments and confirmation, or Bitcoin address (e.g. 1N517iag9jJgTHD1VXjvLCEnZuQ3rJED9L) if recipient is not online.

Pay To: 192.168.0.12 Paste Address Book...

Amount: 9.95

From: Bob

Message: Order #12345

Send Cancel

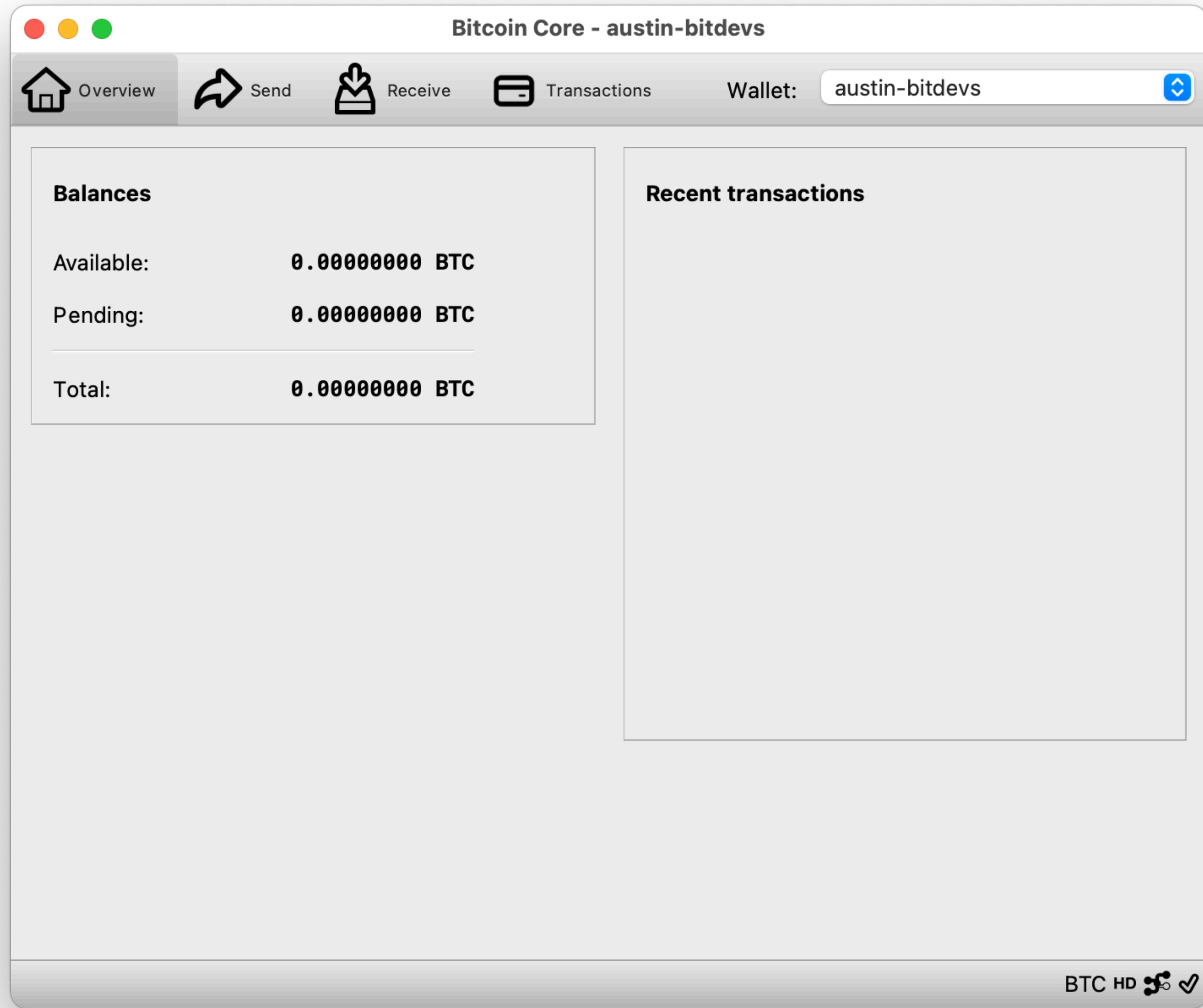
Windows NT/2000/XP/Vista. Open source C++ code is included.

Download link: [bitcoin-0.1.5.rar](http://bitcoin.org/0.1.5.rar)

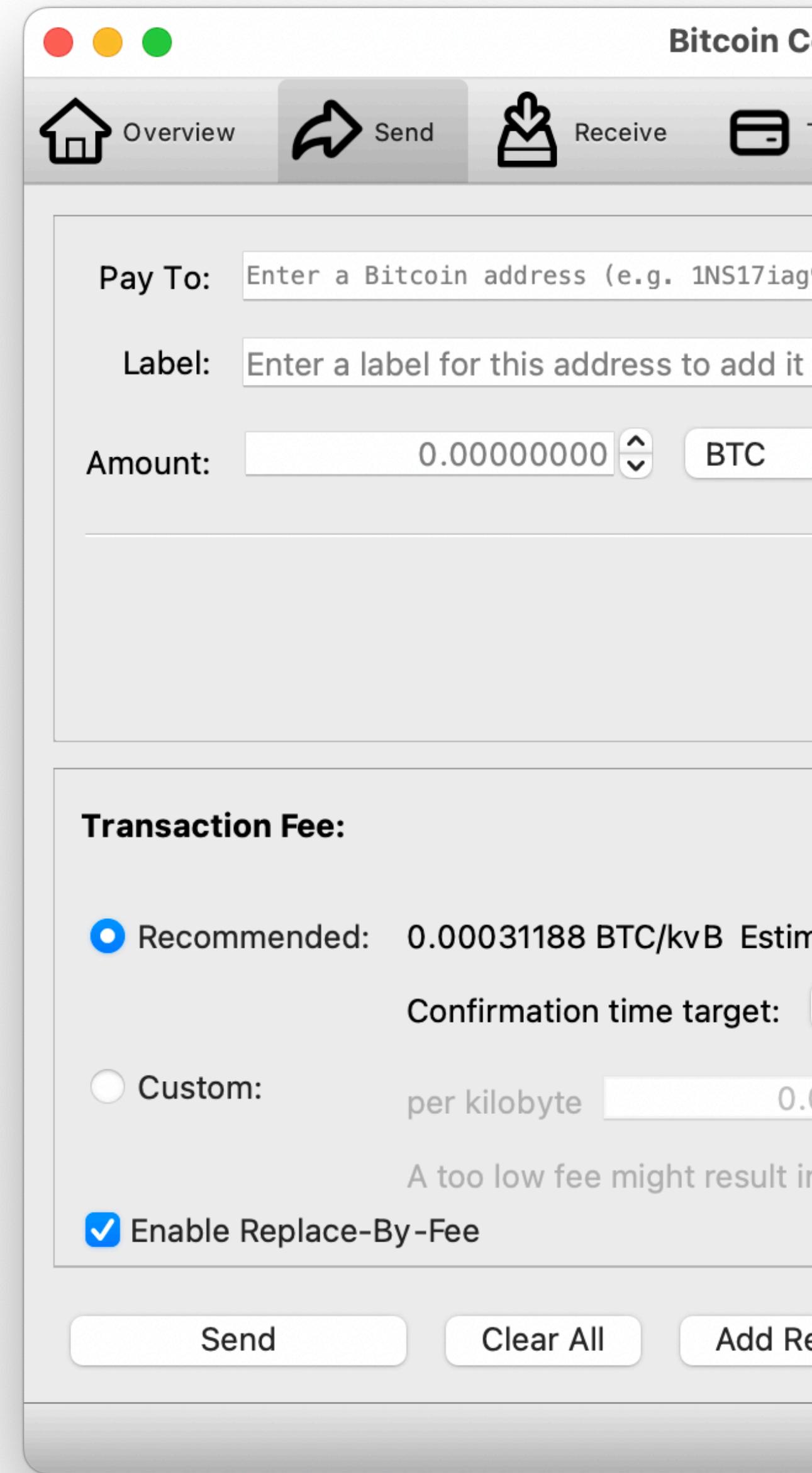
<https://web.archive.org/web/20090303195936/http://bitcoin.org:80/>

<https://github.com/bitcoin-core/gui>

2012



TODAY



github.com

bitcoin-core/gui-qml: Bitcoin GUI (experimental QML-based fork)

bitcoin-core / gui-qml

Type ⌘ to search

Code Issues 33 Pull requests 11 Actions Security Insights

gui-qml Public Edit Pins Unwatch 18 Fork 42 Star 95

main 1 Branch 1 Tags Go to file + Code

hebasto Merge #392: UI Only Custom Datadir c3fb557 · 2 days ago 39,396 Commits

Commit	Message	Time
.github	doc: update pr template	5 months ago
.tx	qt: Bump Transifex slug for 25.x	last year
build-aux/m4	qml: statically link QtQuick2Dialog and FolderLis...	2 weeks ago
build_msvc	Remove unused raw-pointer read helper from un...	9 months ago
ci	ci, gha: Build macOS and Windows artifacts	7 months ago
contrib	build, qt: Enable OpenGL for Windows static buil...	8 months ago
depends	qml: fix file location path in Controls plugin for A...	2 weeks ago
doc	doc: Release notes for testnet defaulting to -acc...	8 months ago
share	depends: Bump MacOS minimum runtime requir...	10 months ago
src	Merge #392: UI Only Custom Datadir	2 days ago

About

Bitcoin GUI (experimental QML-based fork)

Readme

MIT license

Security policy

Activity

Custom properties

95 stars

18 watching

42 forks

Report repository

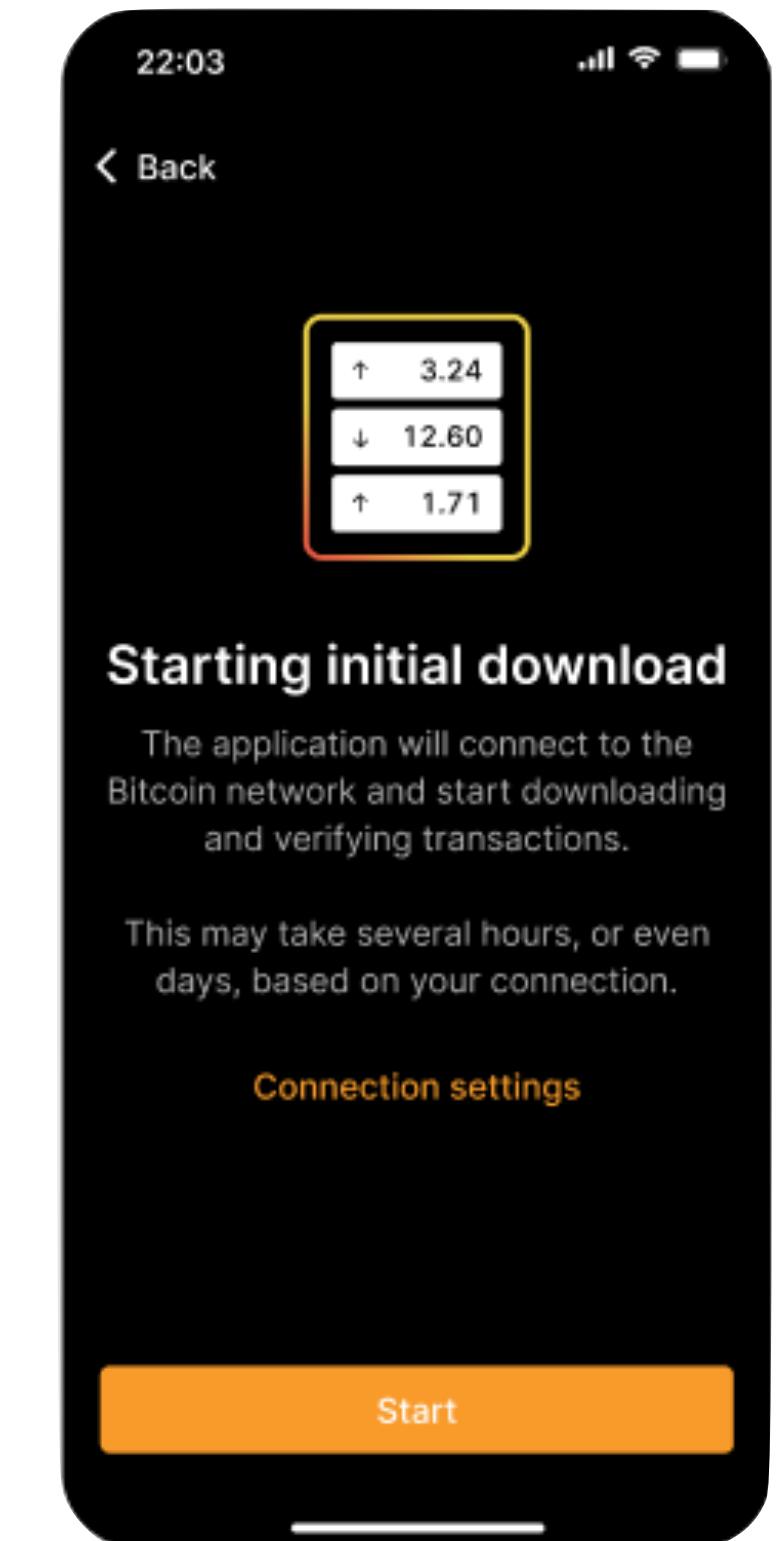
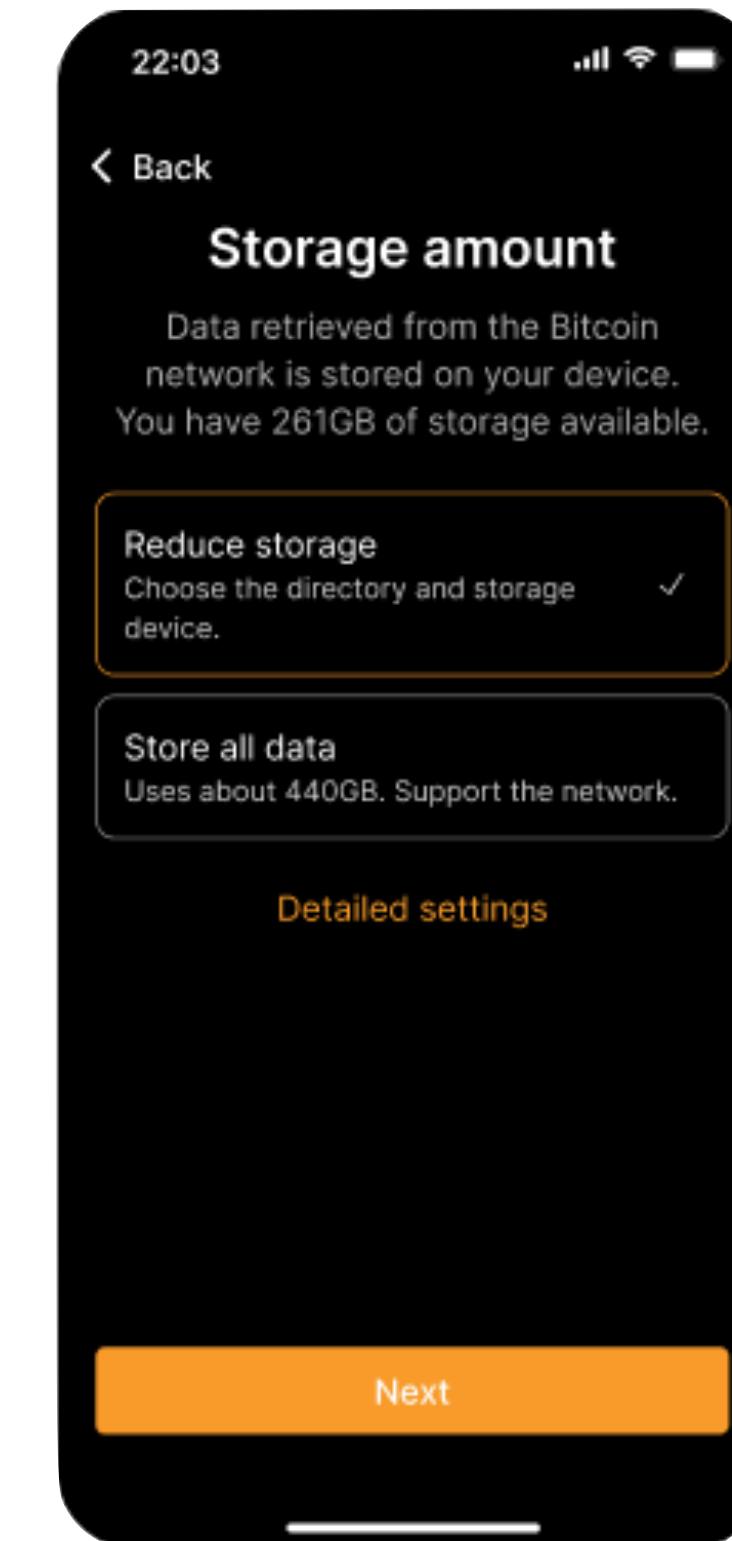
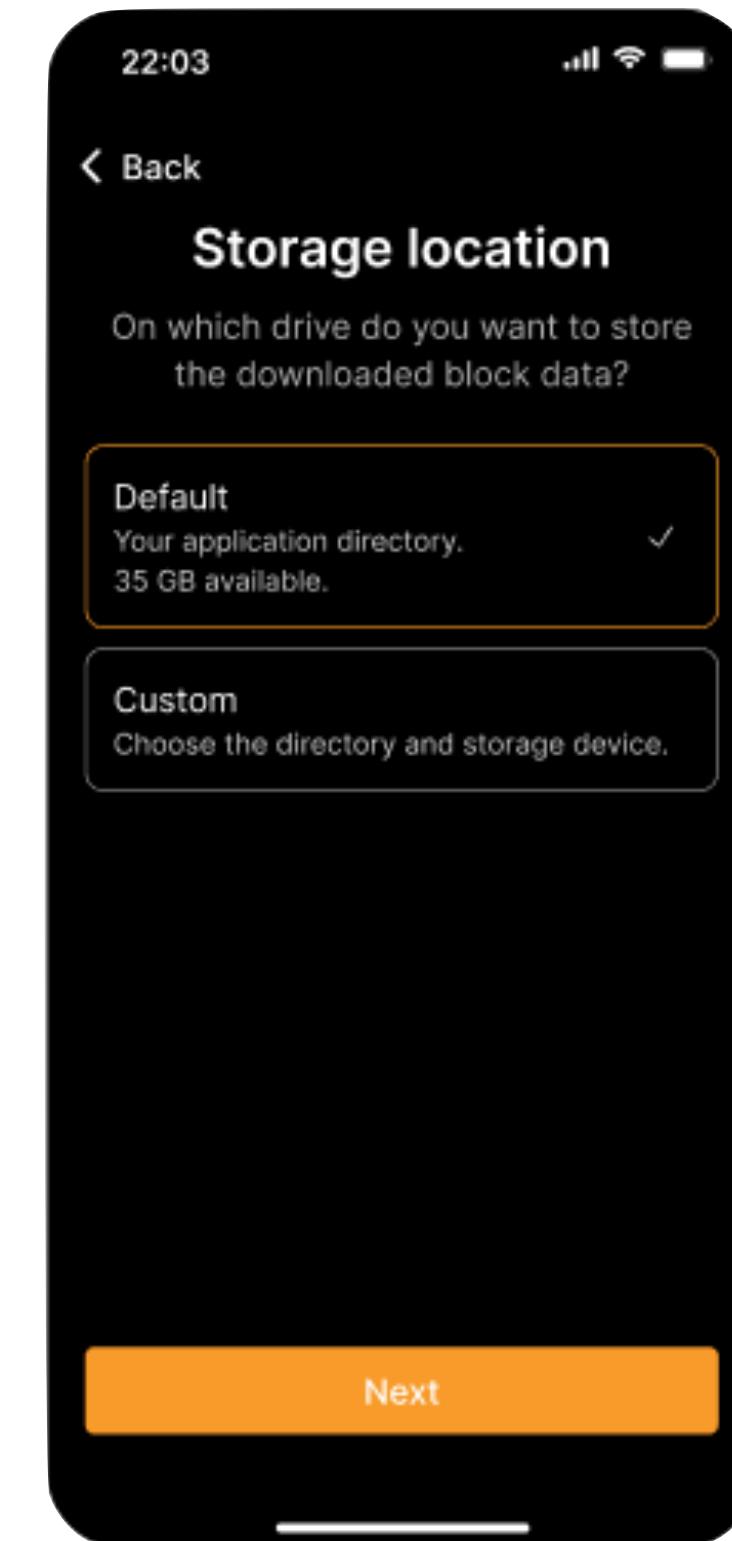
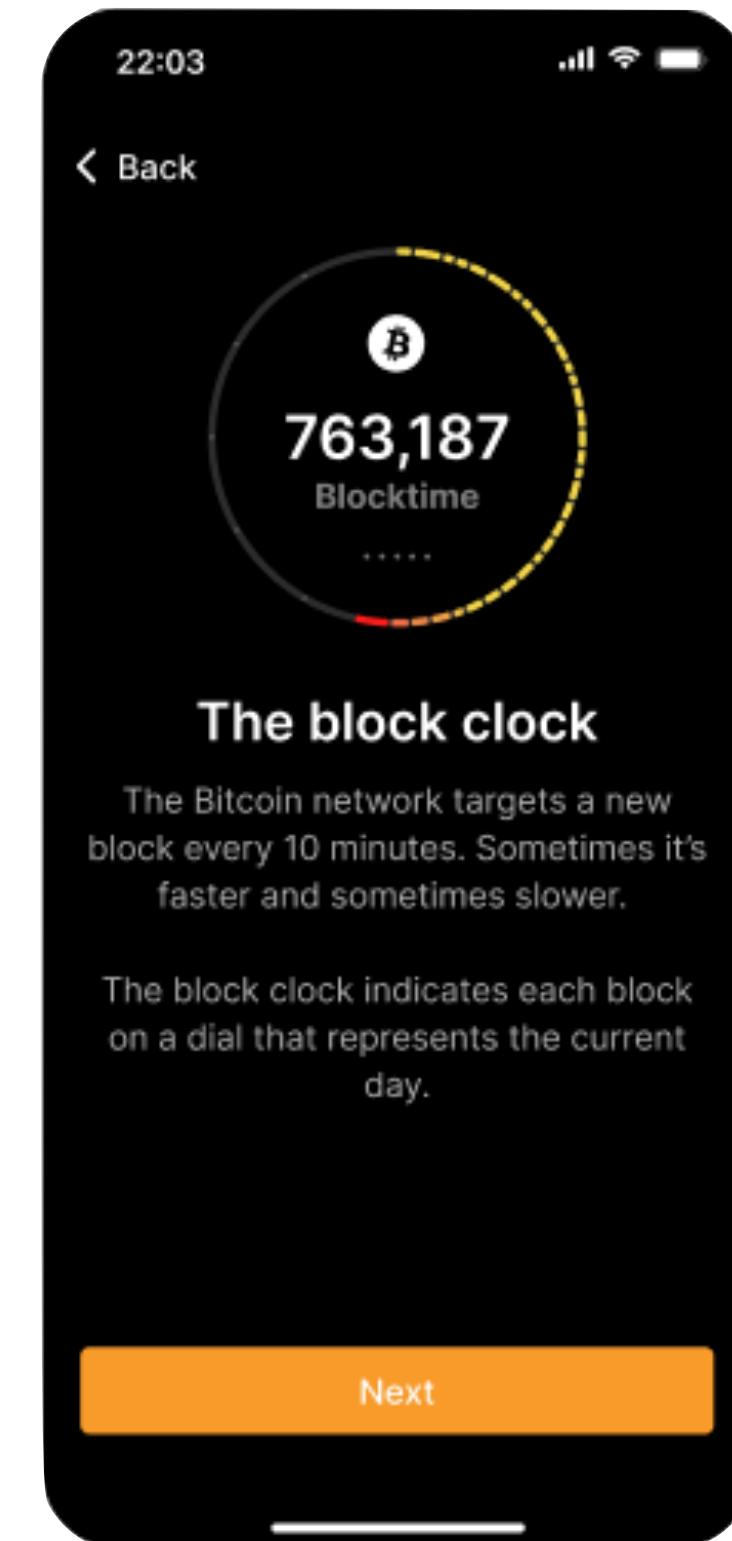
Releases

1 tags

Packages

<https://github.com/bitcoin-core/gui-qml>

<https://github.com/bitcoin-core/gui-qml>



The `bitcoin-core/gui-qml` project seeks to rework and redesign the GUI shipped with Bitcoin Core from the ground up.

Why?

Why put effort into the  
GUI?

Why rewrite/rework the GUI?

Why put effort into the  
GUI?

Why rewrite/rework the GUI?

It allows for normal every day individuals to run Bitcoin Core, in a trust minimized way.

It makes real humans a design target for the project.

The project is more interesting and impactful when it's more than just developer tooling, or corporate consumed software.

Why put effort into the  
GUI?

Why rewrite/rework the GUI?

# Why You Should Think Twice Before Rewriting Existing Software

A project rewrite is not always an optimal solution



justindfuller.com

<https://www.justindfuller.com> › 2020/01 › why-do-we... ::

## Why do we fall into the rewrite trap? - Justin Fuller

Jan 21, 2020 — They don't just stop at "Go is better than Java for this problem" — no, they say

**Java is always bad, always the wrong choice**, and our existing ...

Jeremy Burton in coding, software economics, startups, strategy · June 10, 2011 · 2,308 Words

# Why You Should (Almost) Never Rewrite Code – A Graphical Guide

## Five reasons you shouldn't rewrite that code

Building better software

## Don't Rewrite that Code

This is a plea to the perfectionist in all developers — don't rewrite that code!

What does the project need  
from the GUI?

What do users want?

What does the project need  
from the GUI?

What do users want?

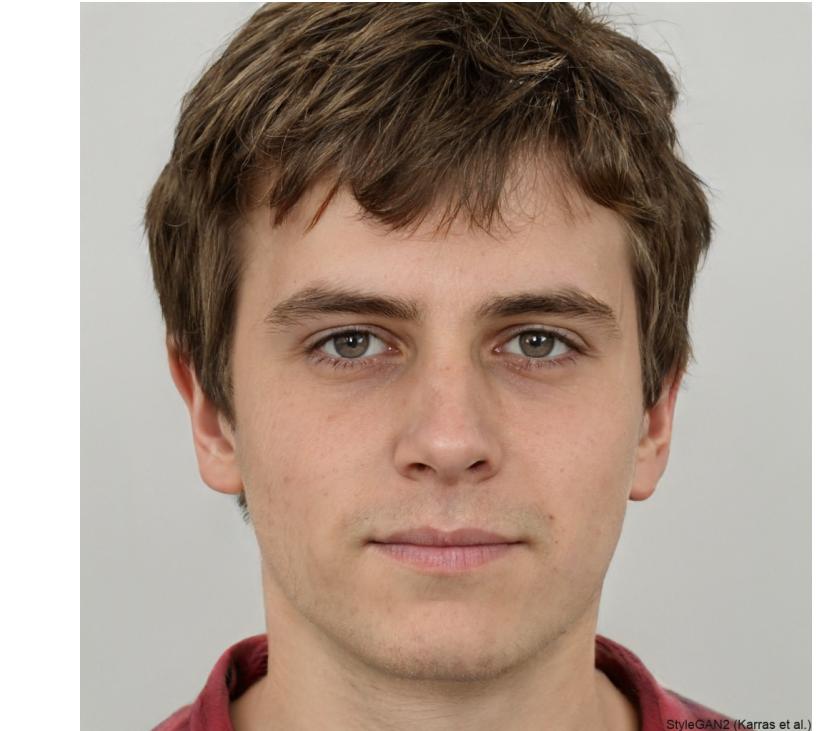
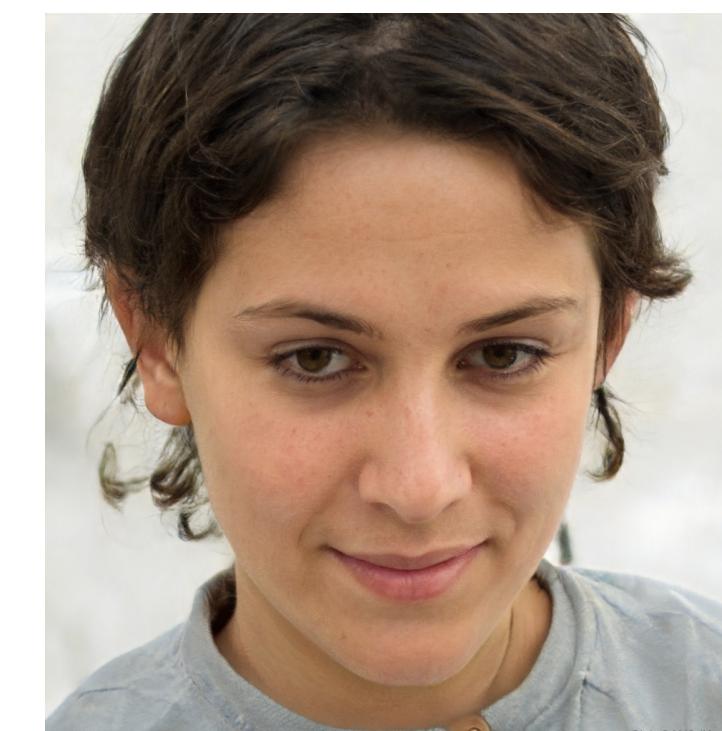
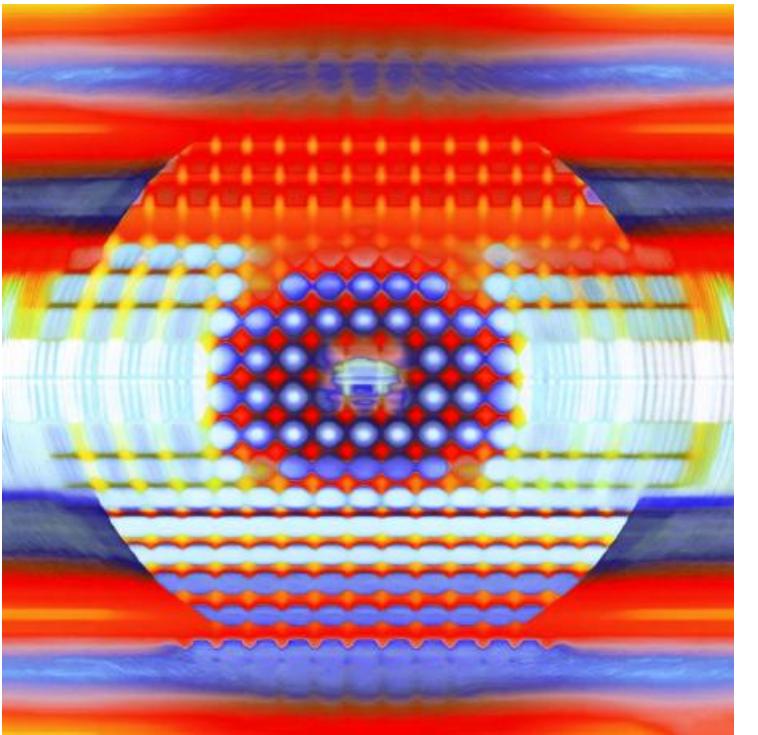
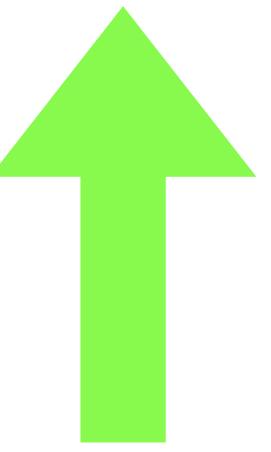
# What does the project need from the GUI?

1. GUI development to be sustainable
2. Not block/slow down other developments in core

# What does the project need from the GUI?

1. GUI development to be sustainable
2. Not block/slow down other developments in core

# Meet the team



# What does the project need from the GUI?

1. GUI development to be sustainable
2. GUI doesn't block/slow down other developments in core

github.com

bitcoin-core / gui

Code Issues Pull requests Projects Security Insights

# Remove confusing "Dust" label from coincontrol / sendcoins dialog #719

Merged hebasto merged 2 commits into bitcoin-core:master from theStack:gui-nuke\_cc\_dust\_label last week

Conversation 17 Commits 2 Checks 15 Files changed 6 +14 -107

Changes from all commits File filter Conversations Review changes

Filter changed files

src/qt/coincontroldialog.cpp

```
@@ -70,23 +70,20 @@ CoinControlDialog::CoinControlDialog(CCoinControl&
coin_control, WalletModel* _m
70 70 QAction *clipboardFeeAction = new QAction(tr("Copy fee"), this);
71 71 QAction *clipboardAfterFeeAction = new QAction(tr("Copy after
fee"), this);
72 72 QAction *clipboardBytesAction = new QAction(tr("Copy bytes"),
this);
73 - QAction *clipboardLowOutputAction = new QAction(tr("Copy dust"),
this);
74 73 QAction *clipboardChangeAction = new QAction(tr("Copy change"),
this).
```



Dust Label



# What does the project need from the GUI?

1. GUI development to be sustainable
2. GUI doesn't block/slow down other developments in core

Merged

p2p: Implement anti-DoS headers sync #25717



Changes from all commits

File filter

Conversations



0 / 55 files viewed

Review changes

pow.h

qt

bitcoin.cpp

bitcoingui.cpp

bitcoingui.h

clientmodel.cpp

clientmodel.h

modaloverlay.cpp

modaloverlay.h

rpcconsole.cpp

rpcconsole.h

sendcoinsdialog.cpp

sendcoinsdialog.h

rpc

minina.cpp

src/qt/clientmodel.cpp

```
18 m_cached_num_blocks = tip.block_height;
WITH_LOCK(m_cached_tip_mutex, m_cached_tip_blocks =
tip.block_hash);
}
// Throttle GUI notifications about (a) blocks during initial sync,
and (b) both blocks and headers during reindex.
- const bool throttle = (sync_state != SynchronizationState::POST_INIT
&& !header) || sync_state == SynchronizationState::INIT_REINDEX;
+ const bool throttle = (sync_state != SynchronizationState::POST_INIT
&& syncType == SyncType::BLOCK_SYNC) || sync_state ==
SynchronizationState::INIT_REINDEX;
const int64_t now = throttle ? GetTimeMillis() : 0;
- int64_t& nLastUpdateNotification = header ?
nLastHeaderTipUpdateNotification : nLastBlockTipUpdateNotification;
+ int64_t& nLastUpdateNotification = syncType != SyncType::BLOCK_SYNC ?
nLastHeaderTipUpdateNotification : nLastBlockTipUpdateNotification;
if (throttle && now < nLastUpdateNotification +
count_milliseconds(MODEL_UPDATE_DELAY)) {
return;
}
```

What does the project need  
from the GUI?

What do users want?



# What do users want?

1. Modern UI
2. Performant, no blocking calls on main GUI thread
3. Better HWI integration

# What do users want?

1. Modern UI
2. Performant, no blocking calls on main GUI thread
3. Better HWI integration

1. The GUI needs to occupy a minimal footprint code-wise, but is unable to do so under Qt Widgets.
2. The GUI doesn't have enough contributors.
3. While the technical locks can be improved, a redesign of the GUI is extremely labor intensive under Qt Widgets.

Why?

Why put effort into the  
GUI?

Why rewrite/rework the GUI?

The QML project is reworking the GUI in order to:

- Utilize a framework that allows the GUI to occupy a smaller footprint within the project
- Breath new life into the GUI with new devs
- Provide a modern and intuitive UX experience

The QML project is reworking the GUI in order to:

- Utilize a framework that allows the GUI to occupy a smaller footprint within the project
- Breath new life into the GUI with new devs
- Provide a modern and intuitive UX experience

Qt 6.7

[Reference](#)[Getting Started](#)[Overviews](#)Qt 6.7 > Qt QML > [Qt Qml](#) 

# Qt Qml

The Qt Qml module provides a framework for developing applications and libraries with the [QML language](#). It defines and implements the language and engine infrastructure, and provides an API to enable application developers to [register](#) custom QML types and modules and integrate QML code with JavaScript and C++. The Qt Qml module provides both a [QML API](#) and a [C++ API](#).

## Using the Module

github.com

bitcoin-core / gui

Code Issues Pull requests Projects Security Insights

# Remove confusing "Dust" label from coincontrol / sendcoins dialog #719

Merged hebasto merged 2 commits into bitcoin-core:master from theStack:gui-nuke\_cc\_dust\_label last week

Conversation 17 Commits 2 Checks 15 Files changed 6 +14 -107

Changes from all commits File filter Conversations Review changes

Filter changed files

src/qt/coincontroldialog.cpp

```
@@ -70,23 +70,20 @@ CoinControlDialog::CoinControlDialog(CCoinControl&
coin_control, WalletModel* _m
70 70 QAction *clipboardFeeAction = new QAction(tr("Copy fee"), this);
71 71 QAction *clipboardAfterFeeAction = new QAction(tr("Copy after
fee"), this);
72 72 QAction *clipboardBytesAction = new QAction(tr("Copy bytes"),
this);
73 - QAction *clipboardLowOutputAction = new QAction(tr("Copy dust"),
this);
74 73 QAction *clipboardChangeAction = new QAction(tr("Copy change"),
this).
```



Dust Label

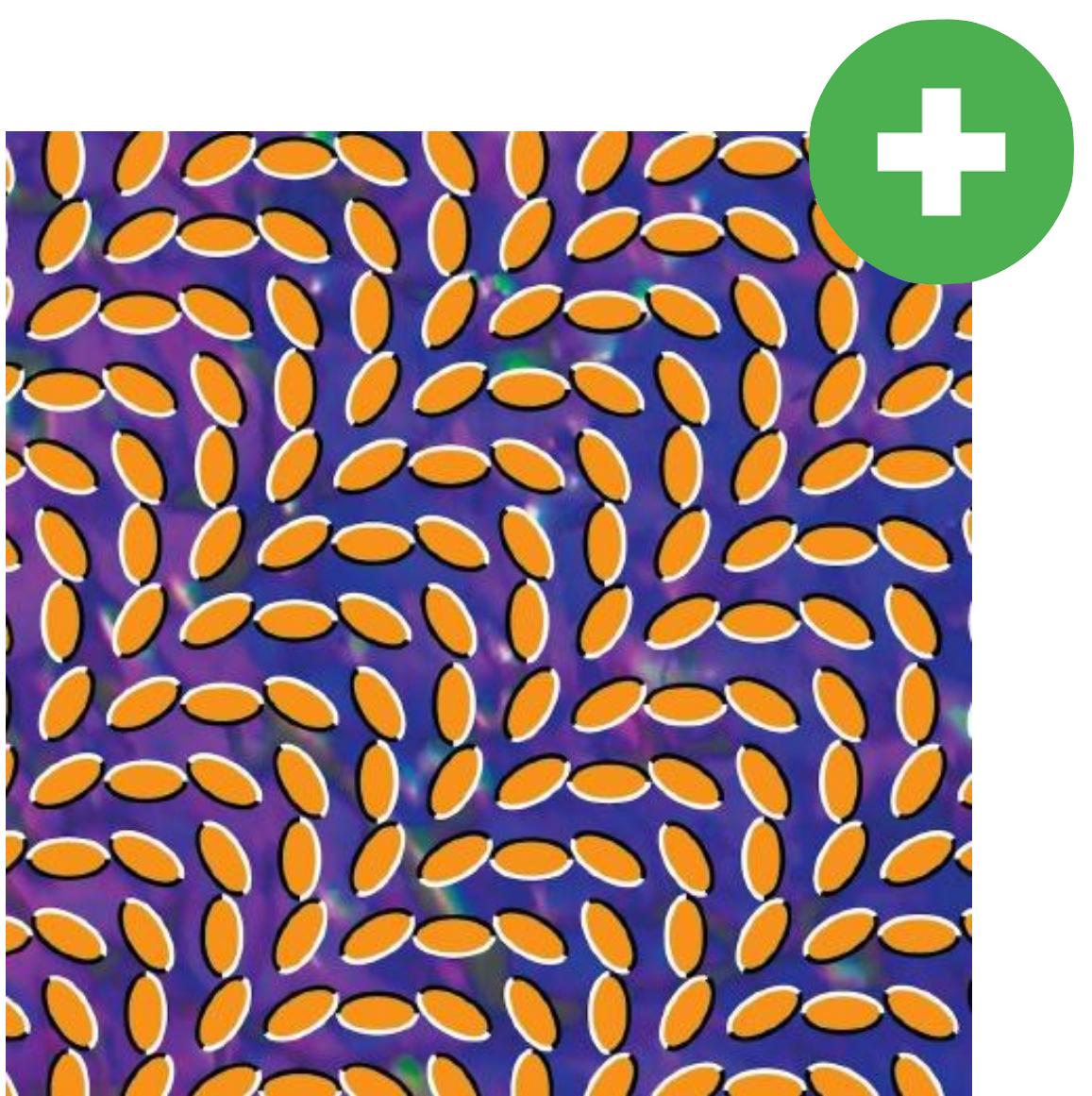
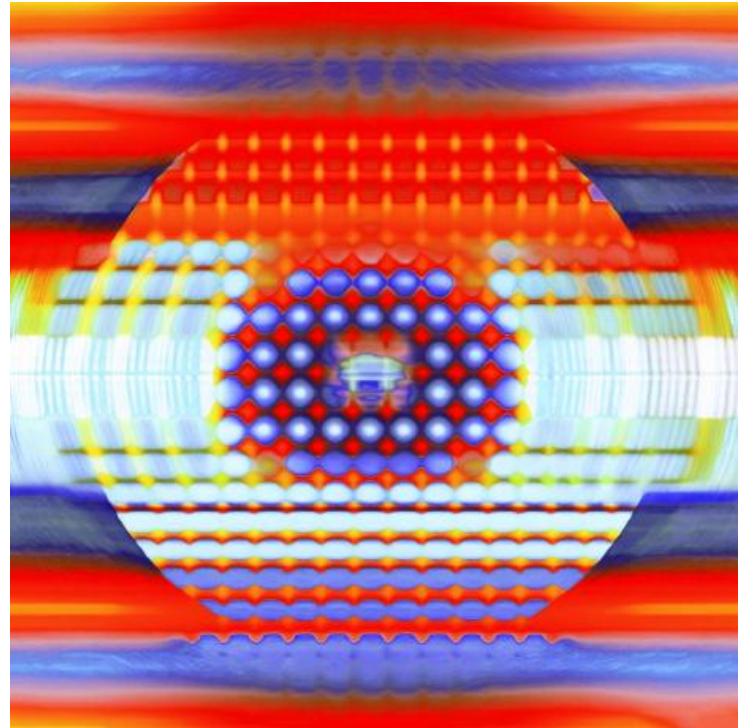




Dust Label

The QML project is reworking the GUI in order to:

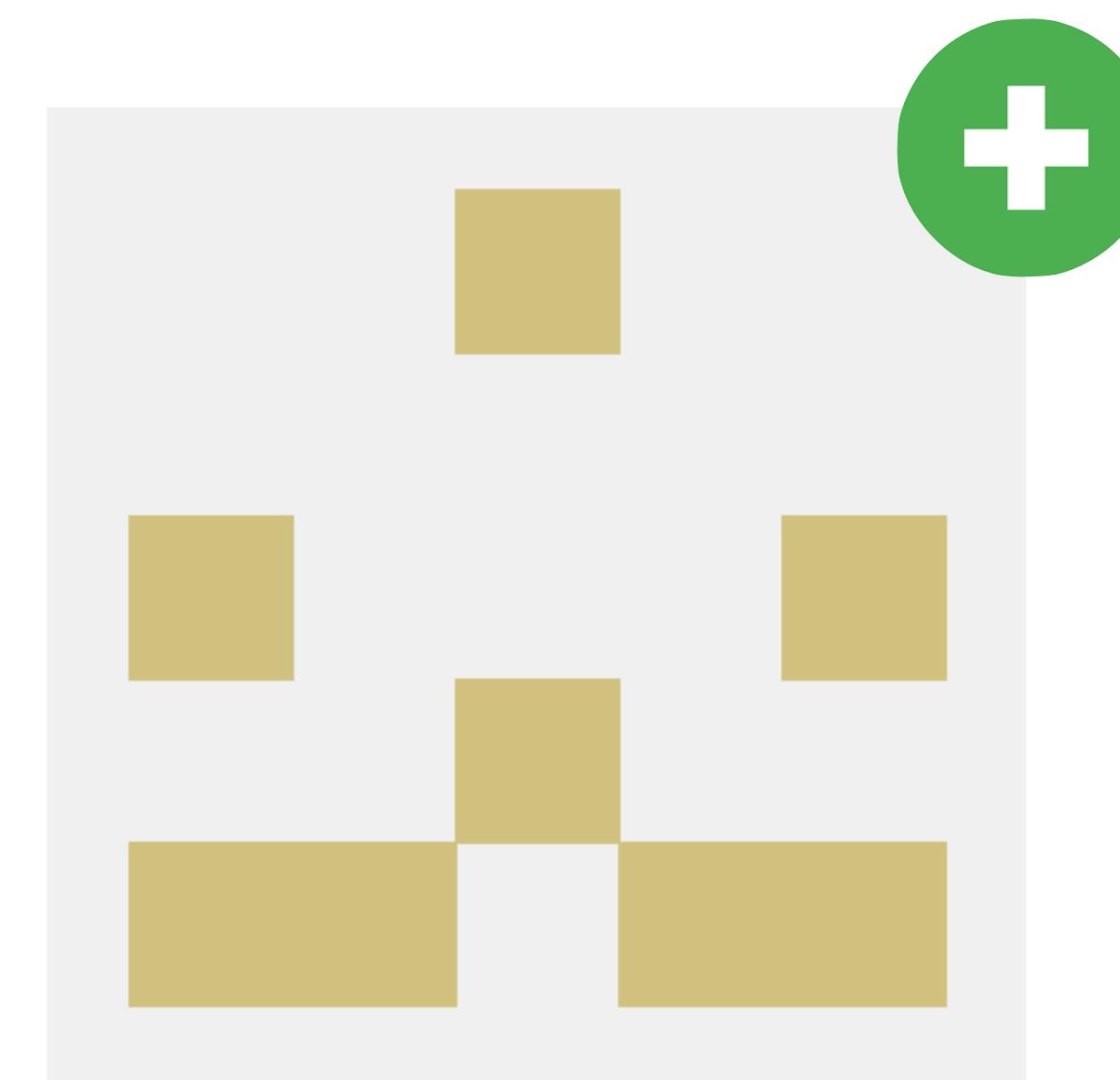
- Utilize a framework that allows the GUI to occupy a smaller footprint within the project
- Breath new life into the GUI with new devs
- Provide a modern and intuitive UX experience



[@johnny9](#)



[@D33r-Gee](#)



[@pablomartin4btc](#)

The QML project is reworking the GUI in order to:

- Utilize a framework that allows the GUI to occupy a smaller footprint within the project
- Breath new life into the GUI with new devs
- Provide a modern and intuitive UX experience



# Design bitcoin for everyone

Open-source design for bitcoin products.

[Join us on Discord](#)[Design guide](#)

Header design by [Rama Krushna](#)

[Submit your own](#) [Next](#)

## We are helping make bitcoin more intuitive and accessible.

As bitcoin's popularity continues to rise, it is essential that everyone be able to participate in this new economy regardless of technical expertise or geography. That can only happen

## Design system

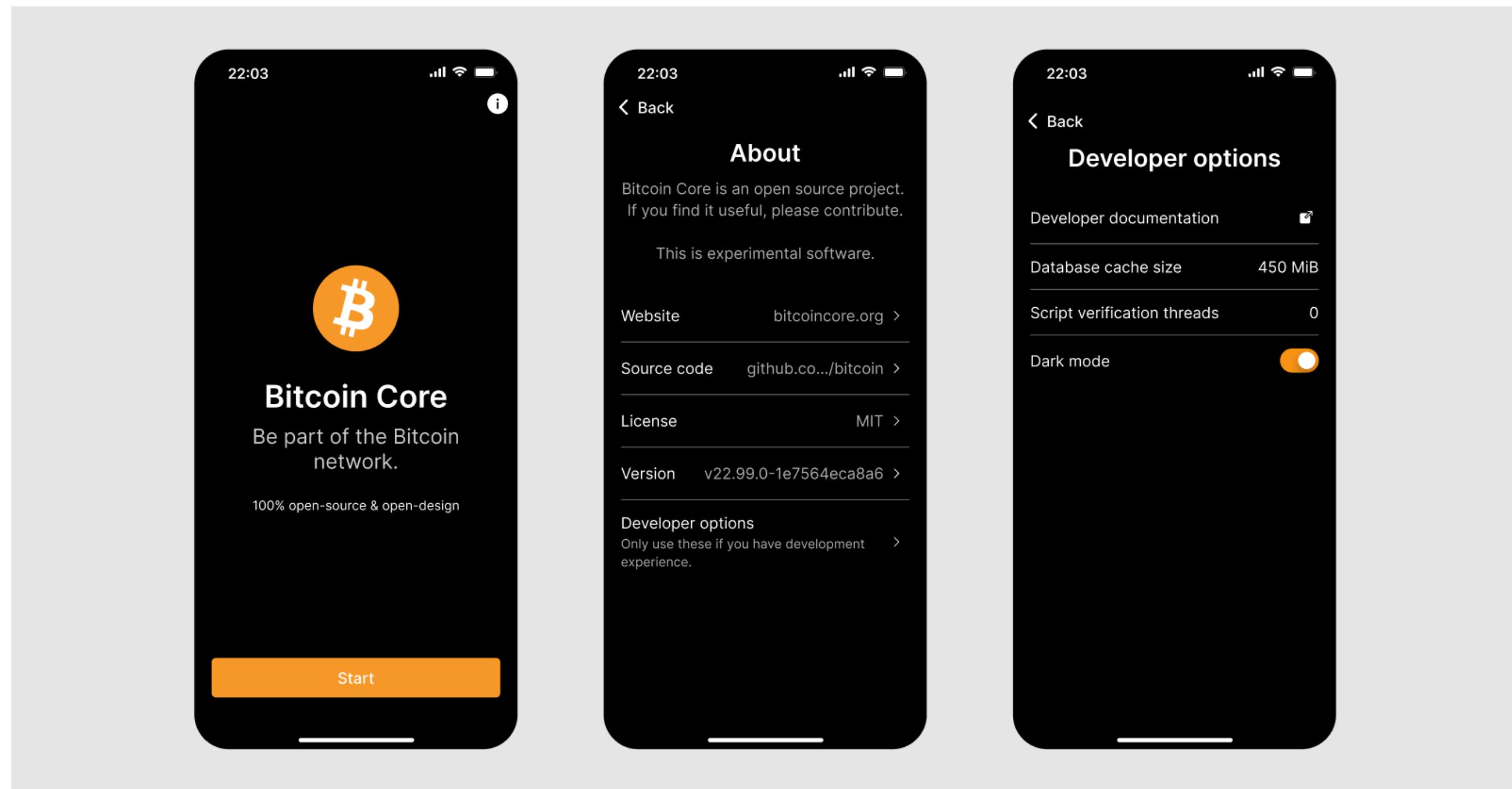
## Milestone 1

## Future milestones

# First use

The first time a user tries an application is an essential experience that can make them appreciate the features and happily return for repeat use, or it can turn them away forever. The reality is usually somewhere in between and depends on the user and their context. But, with a well-considered first-use experience, we can help them get a good start.

The cover screen also offers access to the about screen and developer options.



Ideally, users already clearly understand the application benefits and unique features. The website from which it is downloaded should already have communicated these. However, it can still be beneficial to repeat them. This clears up false expectations and prepares users for what they are about to do next.

# Colors

The color scheme is intentionally kept simple. Generally, there are 10 neutral shades that are used for the majority of elements, and 5 colors used for highlights. Minor variations have been added for specific purposes like interactive states and unique components.

The scheme has two variations, for dark and light modes.

Light Dark

## Primary & accents

Orange is the primary color used to indicate interactive elements. Orange light 1 & 2 are used for interaction states (hover, press). The other colors are used more freely.



Orange

#F7931A



Orange light 1

#FFAD4A



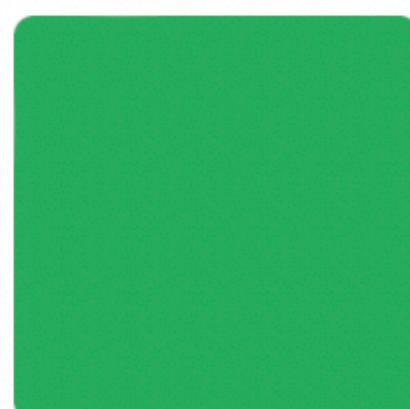
Orange light 2

#FFAD4A



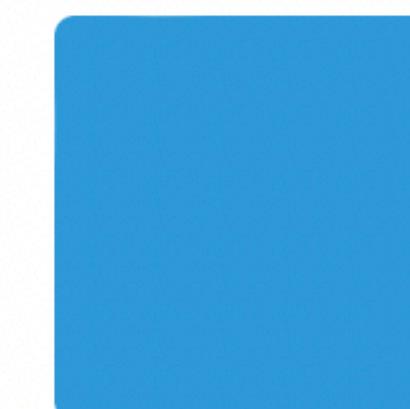
Red

#F7931A



Green

#F7931A



Blue

#F7931A



Purple

#F7931A

## Pages

## Introduction

-

V1: Node runner

V2: Wallet

Peers screen

-

Design system

Icons

Nav components

Design docs website

-

## Frame 2

# □ bitcoin-core-app-cover



# Bitcoin Core App

Frame 2

## Introduction

This is the main design source file for the Bitcoin Core App. Before diving in here, make sure to check out the [full design documentation](#).

Introduction



✓ V1: Node runner

V2: Wallet

Peers screen

Design system

Frame 9840

Block clock

Block clock

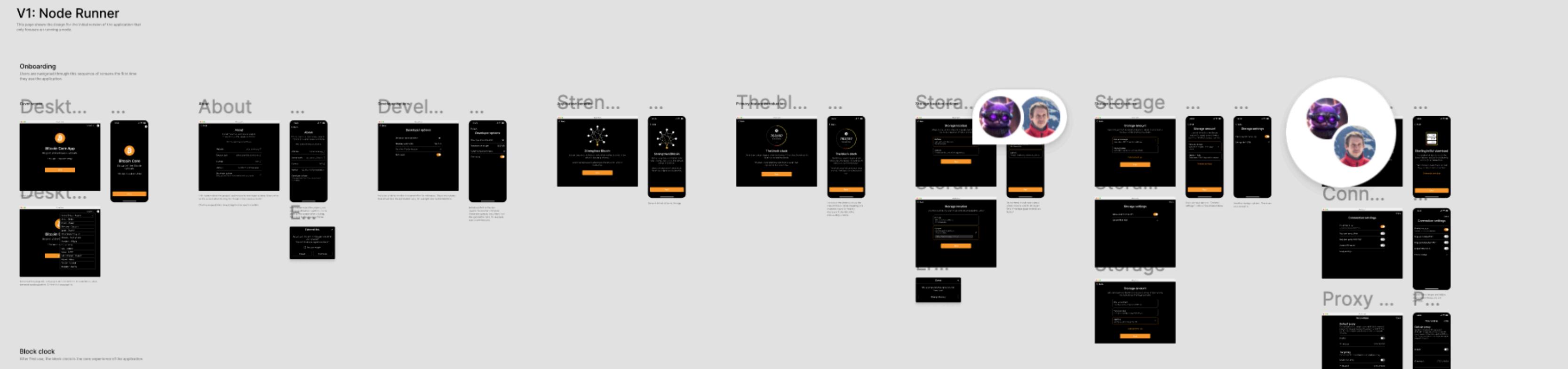
External link modal

Error modal

Block clock

Block clock

Bitcoin Core App desktop layout s...



Proxy settings

Connection settings

Storage settings

Settings

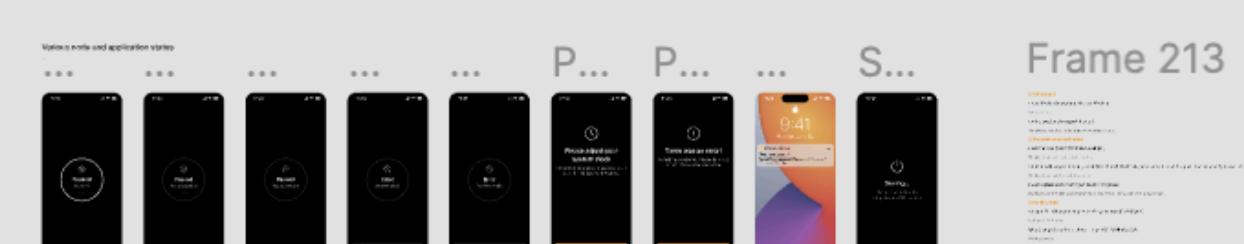
Storage

Storage

Storage location

Storage location

The block clock



Bitcoin Core app design docs | Design documentation for the Bitcoin Core QML application.

# Introduction

These are work-in-progress design documents for the QML-based Bitcoin Core application that is currently in early development. The goal of this site is to document and make easily accessible various considerations and decisions that went into the design of the user interface and user experience.

- [Project planning board](#)
- [Bitcoin Core website](#)
- [Design repo](#)
- [Current design tasks](#)
- [Figma community file \(working file, previous\)](#)
- [Code repository](#)
- [Web prototype](#)

## Provide early feedback

You are invited to be an early tester of the application and provide your feedback to help us in creating a great experience. Keep in mind that the application is experimental, you are installing and running it at your own risk.

The current application only includes a simple [onboarding process](#) and the [block clock](#). It does **not** include wallet features (those will be added in [future milestones](#)). It is available for Linux, Windows, MacOS, and Android operating systems.



## Milestones 1.x

### 1.2 Navigation

### 1.3 Import

### 1.4 Create

### 1.5 Activity

### 1.6 Receive

### 1.7 Send

### 1.8 Other

Explorer

Console

Showcase

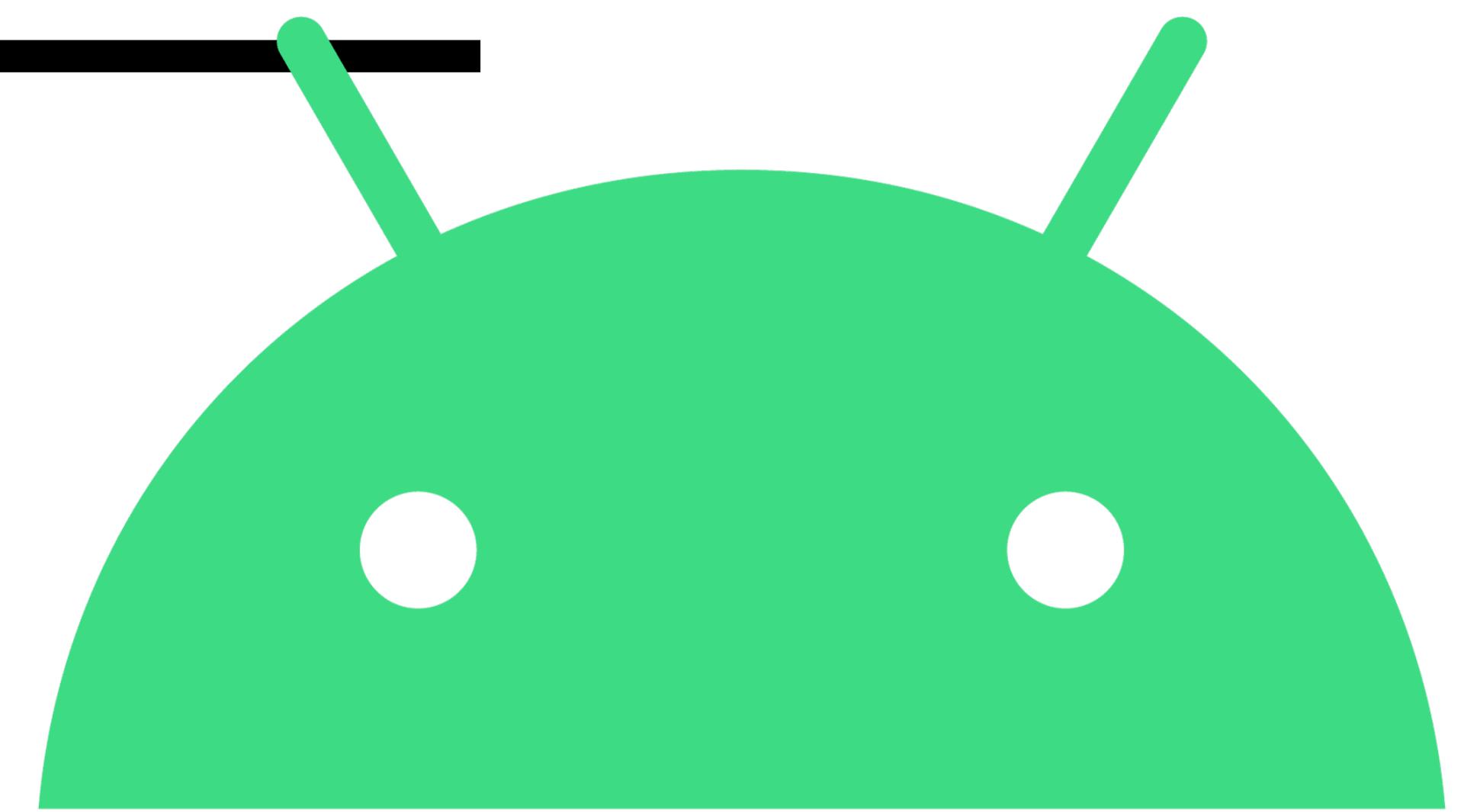


## Feedback we are looking for right now

- Did you install it on a phone, tablet or desktop?
- What is the device brand and model?

# Android

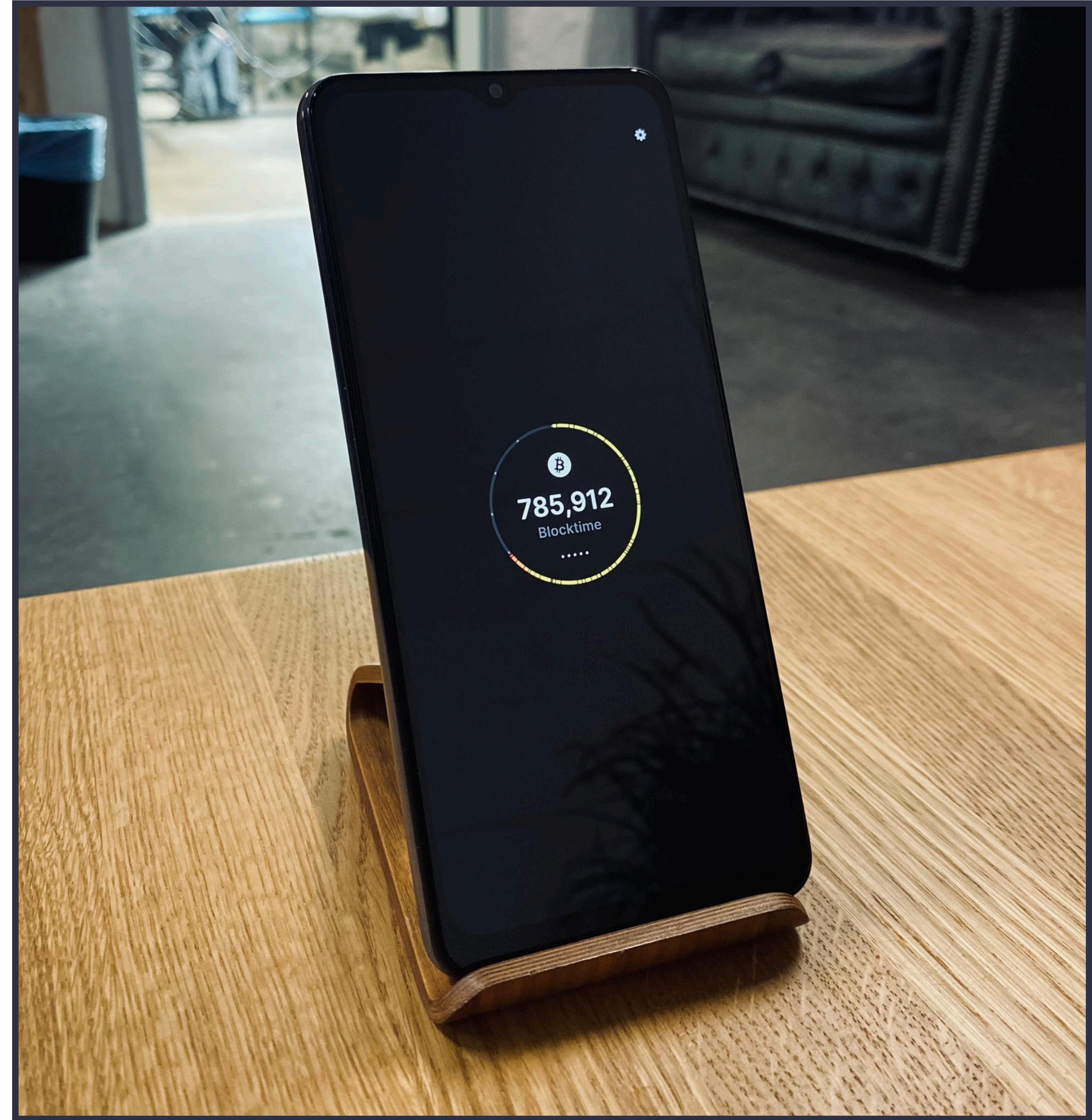
---



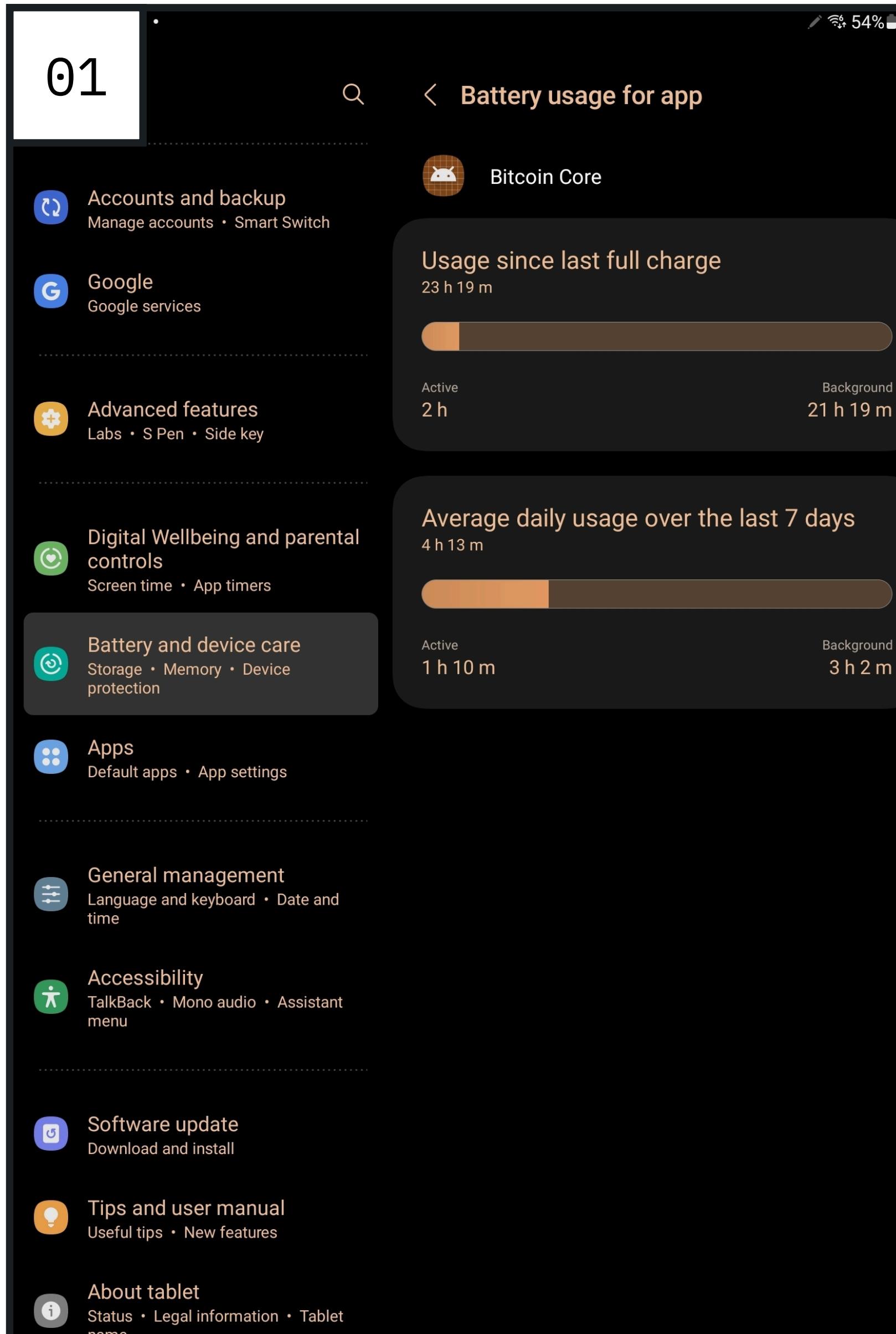


Left: Samsung Galaxy Tab S8

Right: Samsung Galaxy A12



# post-IBD power usage



01

## Settings



### < Battery usage for app

 Accounts and backup  
Manage accounts • Smart Switch

 Google  
Google services

 Advanced features  
Labs • S Pen • Side key

 Digital Wellbeing and parental controls  
Screen time • App timers

 Battery and device care  
Storage • Memory • Device protection



Bitcoin Core

#### Usage since last full charge

23 h 19 m

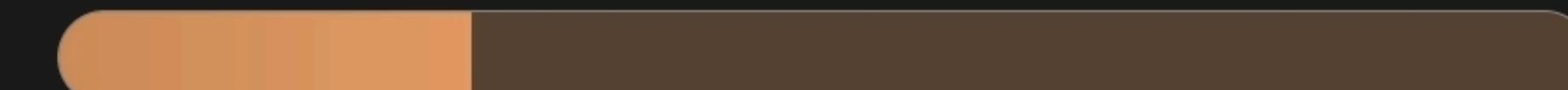


Active  
2 h

Background  
21 h 19 m

#### Average daily usage over the last 7 days

4 h 13 m



Active  
1 h 10 m

Background  
3 h 2 m

02

Google

Google services



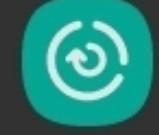
Advanced features

Labs • S Pen • Side key



Digital Wellbeing and parental controls

Screen time • App timers



Battery and device care

Storage • Memory • Device protection

[View details](#)



Apps

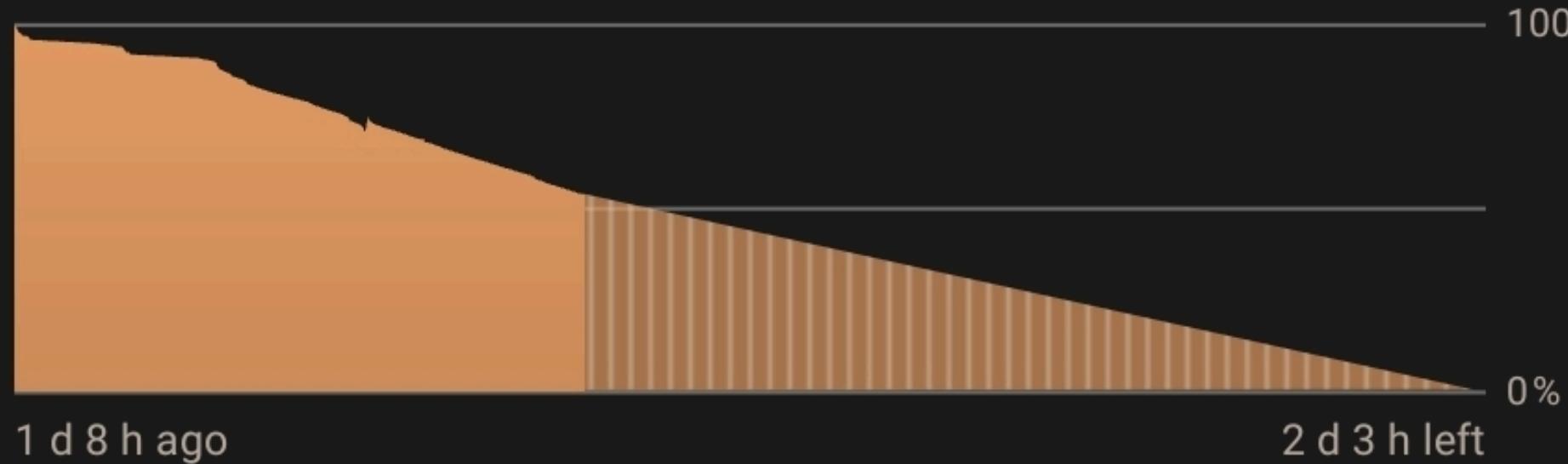
Default apps • App settings



General management

Language and keyboard • Date and time

Usage since last full charge



1 d 8 h ago

2 d 3 h left

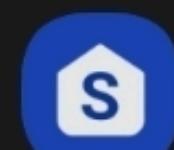
● Battery use  
● Estimated time

A full charge will last about  
3 d 23 h



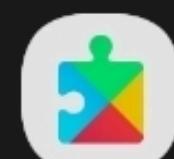
Bitcoin Core

24.7%



One UI Home

1.0%

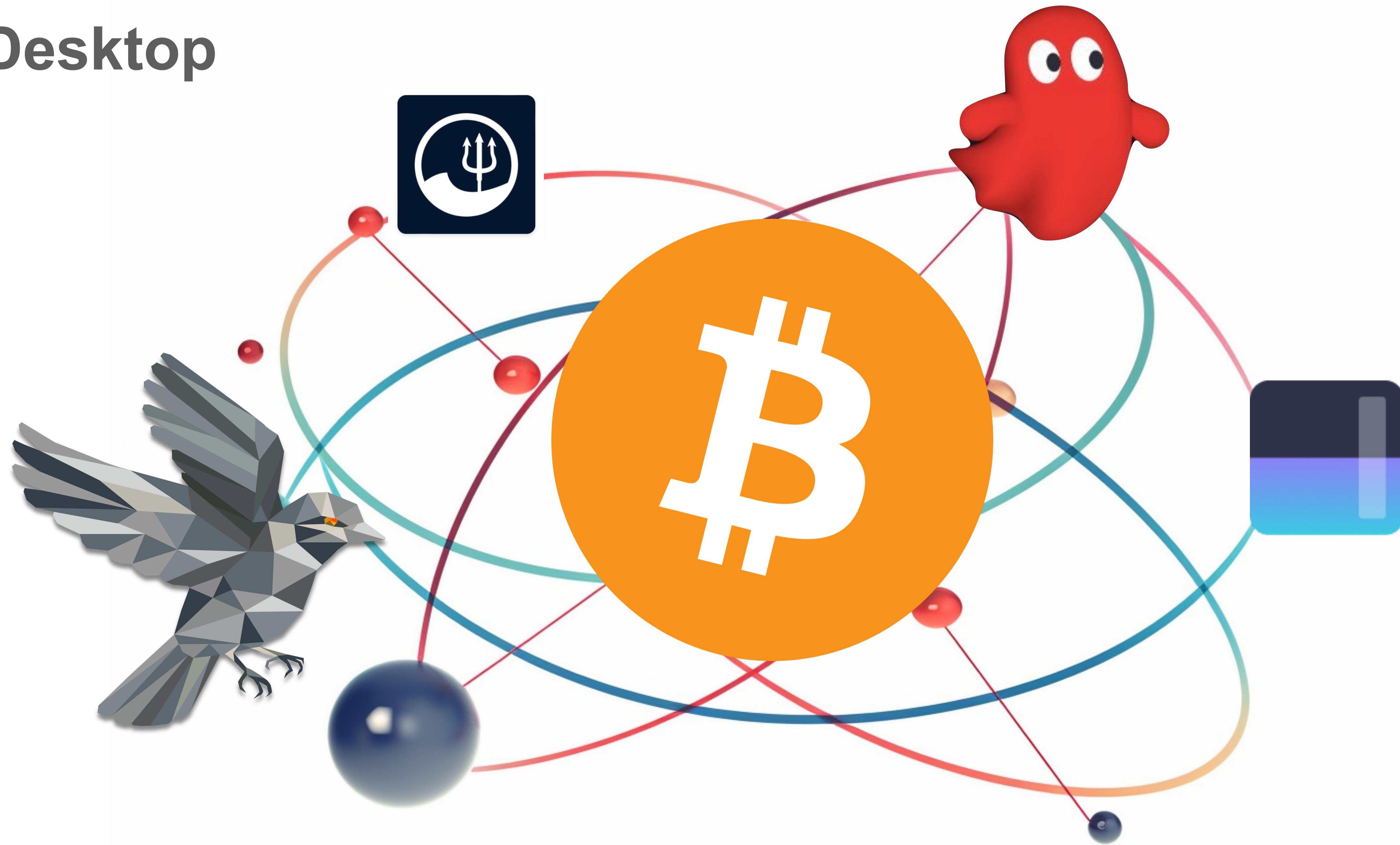


Google Play services

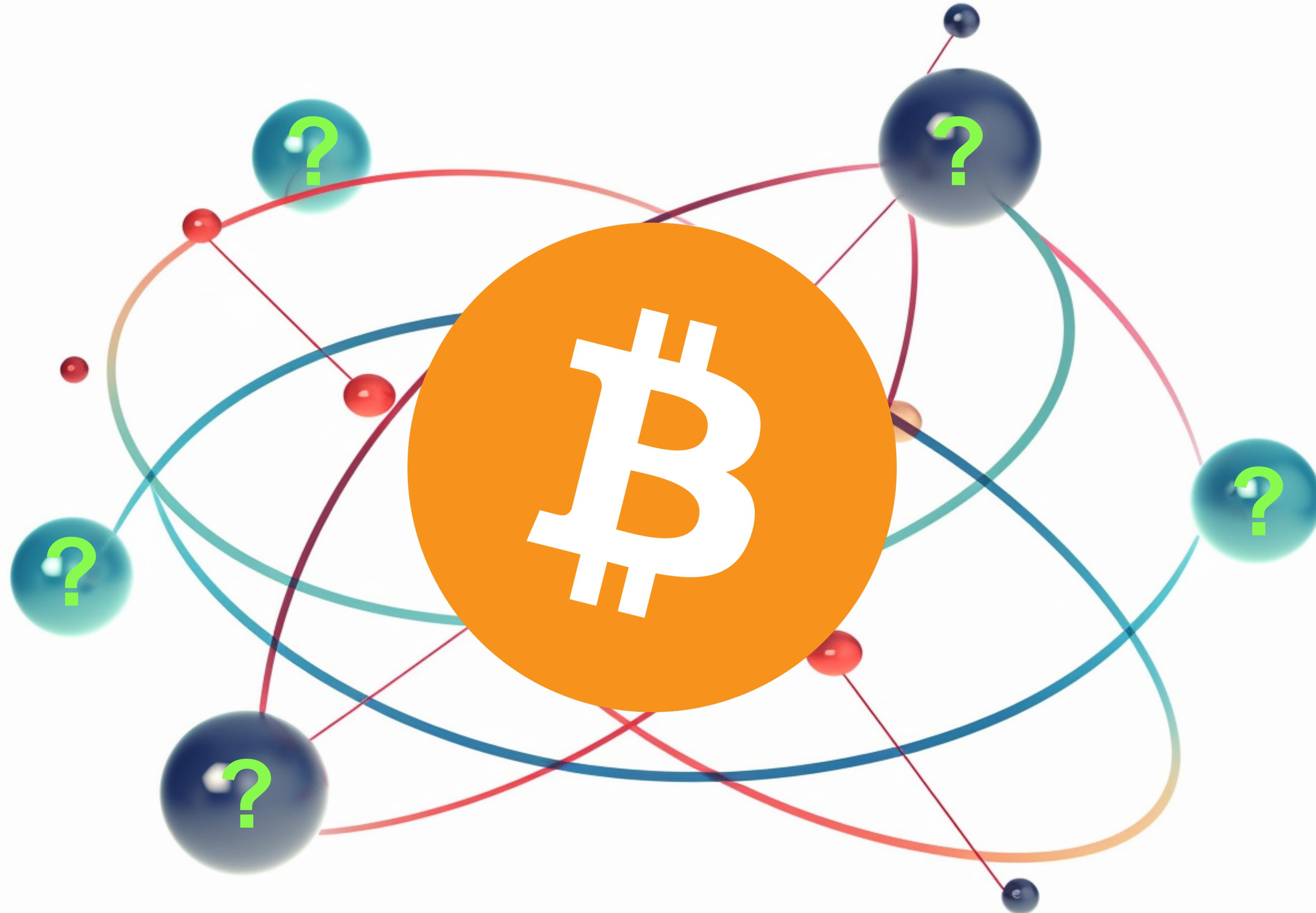
0.3%

• •

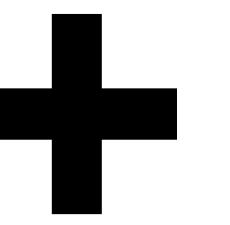
# Desktop



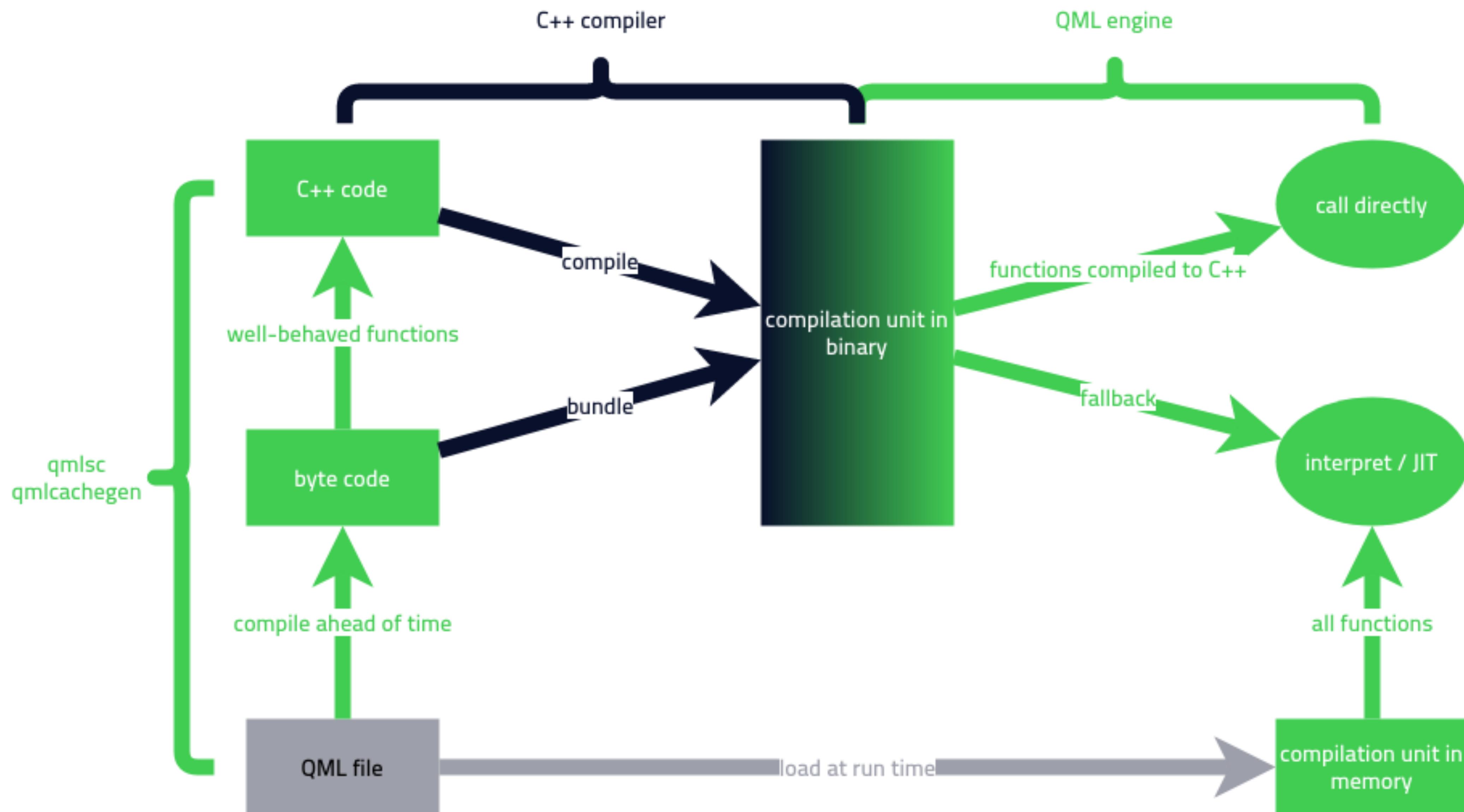
# Mobile



**What's next?**



# New Qt Quick Compiler



1. New, Nice, Sleek QML GUI, less code, smaller footprint
2. New batch of Core Devs focused on the GUI
3. Still a C++ GUI using the Qt dependency
4. Opening up core to a larger user base consisting of individuals