We got tired of operating systems from the 1970s. So we built our own from scratch. We realized that this would work horribly with an internet from the 1970s, so we rebuilt the web too.

The result is Urbit. Urbit brings the connectedness of the cloud to everyone with a simple but powerful idea - The personal cloud computer. Arvo is the functional operating system running on your computer. Urbit is the network of everyone's personal cloud computers, Arvo instances. But why is this system better? See for yourself:

Feature Bullet Points

These would open a video demonstration (30-60 second video), or simple paragraph or two on the feature.

Urbit, a better social network.
(Video of making a neighbor, talk to people in different rooms and/or share files and/or Hoon code.)

Arvo, an instantaneously updating OS.
(Video of downloading a simple applications from a remote ship.)

Clay, a filesystem with builtin version control.
(Video demonstrating merges, the path versions.)

Hoon, a powerful, practically typed functional language.
(Write an update for a broken application, watch it instantaneously hot patch.),
(Edit a super simple curl or network thing.),
(Simple ?(Twitter Dropbox Twilio Facebook) API client demo that tweets or downloads a file or something.)

Urbit

Computing isn't a monolith anymore. So why are we using software and protocols from a time when it was? Urbit is a a better way of computing for today: A distributed network of personal cloud computers, running on an operating system and programming language built for data uniformity and networking. Your computer is your identity and we

So much of what we do today is in the cloud, it's becoming a mess both for developers and users to maintain everyone's accounts, data, privacy, and software versions within towering servers. Urbit simplifies web applications by making them personal again, while keeping all of the convenience cloud applications give in distribution, maintenance, and connectedness. Your identity is your Urbit ship and your

Arvo

Arvo is the event-driven, functional OS which forms the nodes of the Urbit network.It is built on the same event library as node.js (libuv) and is written in Hoon, isolated from Unix. Arvo can update itself and its data and programs indefinitely from its own state. This functional nature also makes Arvo modular, concise (Defined in about 6500 lines), and seamlessly connect to other Arvo machines via a network messaging protocol built upon UDP.

Clay

Clay is Arvo's secure, referentially transparent, and decentralized filesystem. Its immutable nature gives Clay a simple but powerful builtin version control system - Every version of all of your files is automatically remembered by change number, date, and label. Unified under Urbit's global namespace, Clay makes pulling updates and downloading new applications from other Arvo machines effortless.

Hoon

Because programmers don't wear lab coats to make and test hypotheses, we also decided to quit pretending they are computer "scientists." So, we created the functional programming language Hoon with a focus on craftsmanship, practicality, and freedom.

Hoon is a strict, typed, functional programming language on which applications in Urbit are built. Hoon's functionality and standard libraries make it especially good at metaprogramming, marshaling and validating untyped data, and decoding and encoding binary message formats, and constructing elegant type hierarchies. These strengths play to those of Arvo, meaning building web applications in Urbit is helped by the operating system instead of impeded by it. Hoon compiles to the homoiconic combinator algebra Nock, which just means you are coding a beautiful AST. Nock will never change, so those applications will run on every Arvo machine, forever.

Hoon is nothing like any programming language you've seen, but that's good thing. Find out why.

Documentation and further propaganda ensues...