MIRAGE OS

Reconstructing library operating systems

http://openmirage.org/

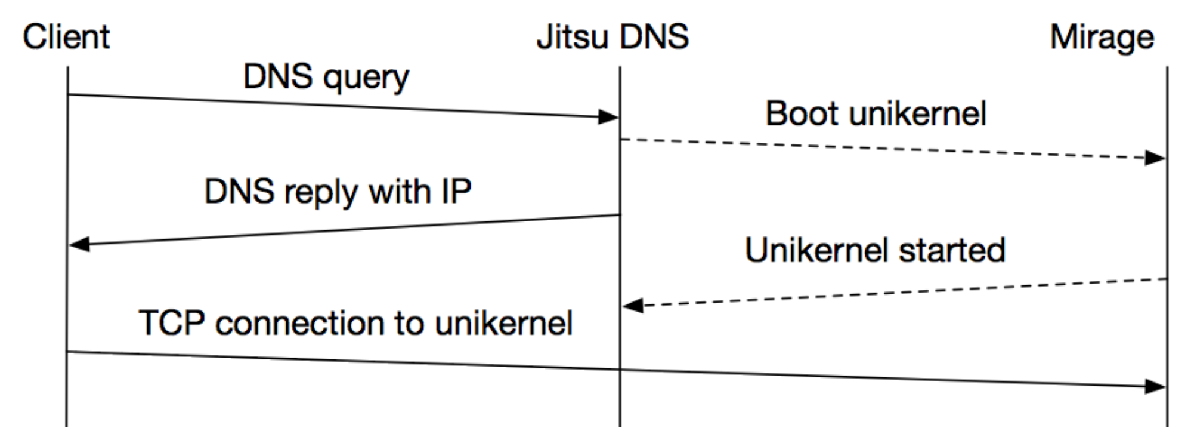
Virtualization is useful in many situations, but adds yet another layer to an already highly-layered software stack now including: support for old physical protocols (e.g. disk standards developed in the 80s such as IDE); irrelevant optimisations (e.g. disk elevator algorithms on SSD drives); backward-compatible interfaces (e.g. POSIX); user-space processes and threads (in addition to VMs on a hypervisor); managed code runtimes (e.g. OCaml, .NET or Java) which all sit beneath your application code.

Are we really doomed to adding new layers of indirection and abstraction every few years, leaving future generations of programmers to become virtual archeologists as they dig through hundreds of layers of software emulation to debug even the simplest applications?

Mirage OS is a **library operating system** that constructs **unikernels** for **secure**, **high-performance** network applications across a variety of **cloud computing** and **mobile** platforms. Code can be developed on a normal OS such as Linux or MacOS X, and then compiled into a fully-standalone, specialised unikernel that runs under the Xen hypervisor.

Mirage is based around the **OCaml** language, with syntax extensions and 50 + libraries which map directly to operating system constructs when being compiled for production deployment. As such, Mirage includes **clean-slate functional implementations** of protocols ranging from **TCP/IP**, **DNS**, **SSH**, **TLS**, **Openflow**, **HTTP**, **XMPP** and **Xen** inter-VM transports.

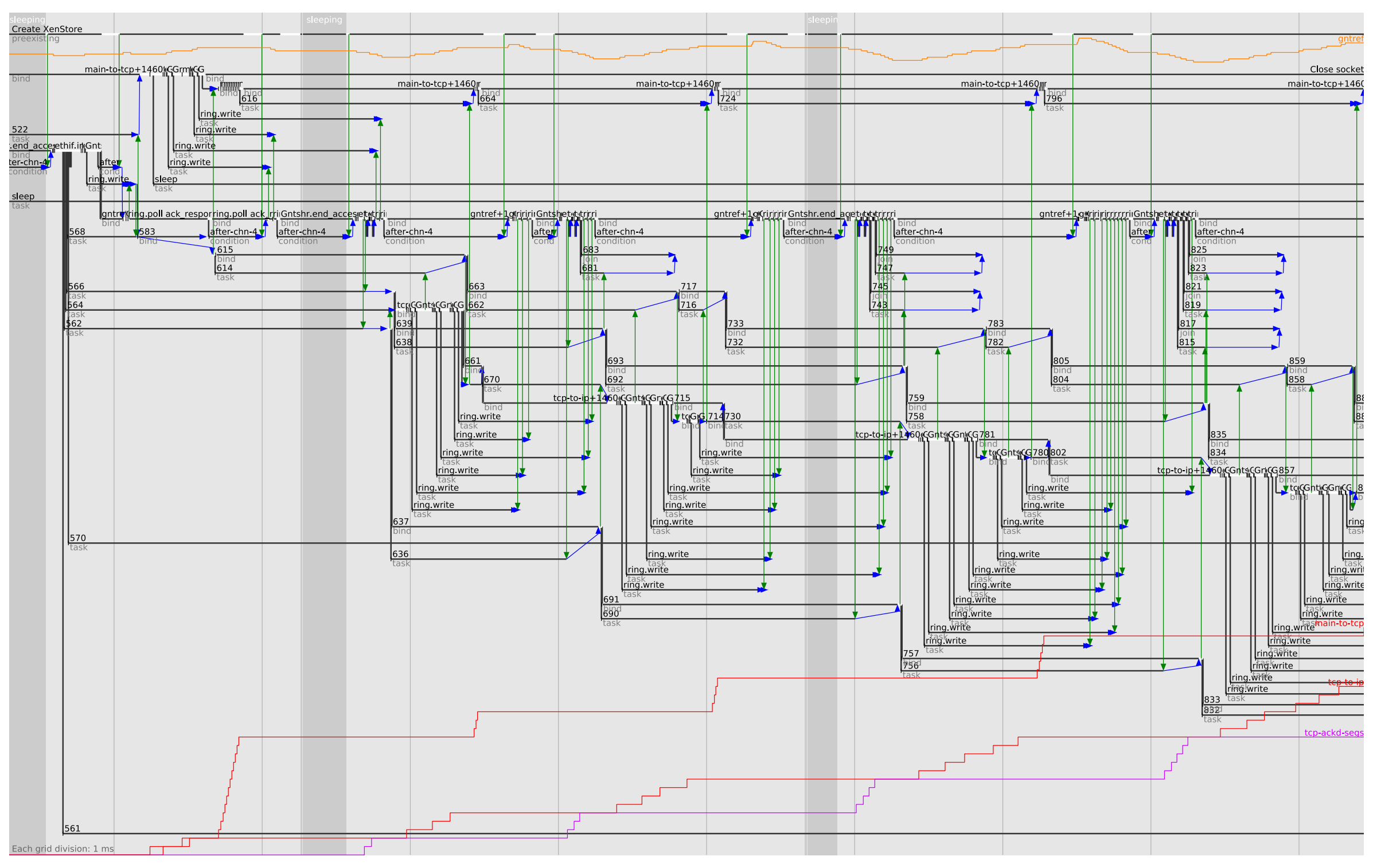
A Mirage web server unikernel running as a Xen guest on an ARM CubieTruck serves up a slide deck.



Example: Jitsu

Just-in-time summoning of unikernels (Jitsu) is a DNS server that saves resources by starting Mirage instances on demand.

Mirage boots in less than 10 ms on x86 and is up and running before the client has received the DNS response and opened a TCP connection.



Example: Tracing

The trace on the left shows a unikernel running on an ARM board transmitting TCP packets.

Horizontal black lines are Lwt threads/promises, green arrows indicate a promise being resolved and blue arrows a result being read.

At the bottom, the three lines show how many TCP packets have been buffered for sending, transmitted, and ack'd by the remote end.

Having everything in a single language makes whole-stack tracing easier than in traditional systems.

UNIVERSITY OF CAMBRIDGE

Technical Contact:
Anil.Madhavapeddy@cl.cam.ac.uk

Nymote

UNIVERSITY OF CAMBRIDGE

Systems Research Group

Amir Chaudhry, Anil Madhavapeddy,
Richard Mortier, Jon Crowcroft
(and many others)

A cohesive set of tools and **software infrastructure** to **build applications** to provide end-users with **life-long control** of their **networks and personal data**.

nymote.org

ocaml.io

Infrastructure

Our approach is to create secure, robust and **open-source** infrastructure that allows us to create distributed systems that anyone can build on. This toolstack must deal with the issues of deployment, sync and connectivity.

Mirage OS

A library OS to create **unikernels** which can use the same code-base to deploy software on both the public cloud and embedded devices, like raspberry pis.

Irmin

A new kind of **library database**, based on the principles of Git, meaning that all history is tracked and can be moved between devices with ease.

Signpost

Improve **end-to-end connectivity** between edge devices without requiring complex configuration. We use DNS updates to provide a constant 'pointer' to your growing number of gadgets.

OCaml

A robust, strongly-typed, **systems programming** language that supports functional, imperative and object-oriented styles.

Applications

With the new infrastructure, we can build robust applications that will push towards more **decentralised** systems. With the right foundations in place, the focus can shift to improving the functionality and extensibility of everyday tools we have come to rely on.

Mail

Make it easy for an end-user to run their own email server. Developers can then write software to interact directly with that server to enhance email in ways we cannot right now.

Contacts

Treat your personal address book as the ultimate social network and make it core to your online interactions. This allows the possibility to incorporate private data to enhance other services without losing ownership.

Calendar

Party invites via social networks and meetings organised at work all need to find their way to one place. With a system under the user's control we can build new features to learn and predict activity without privacy leaks.

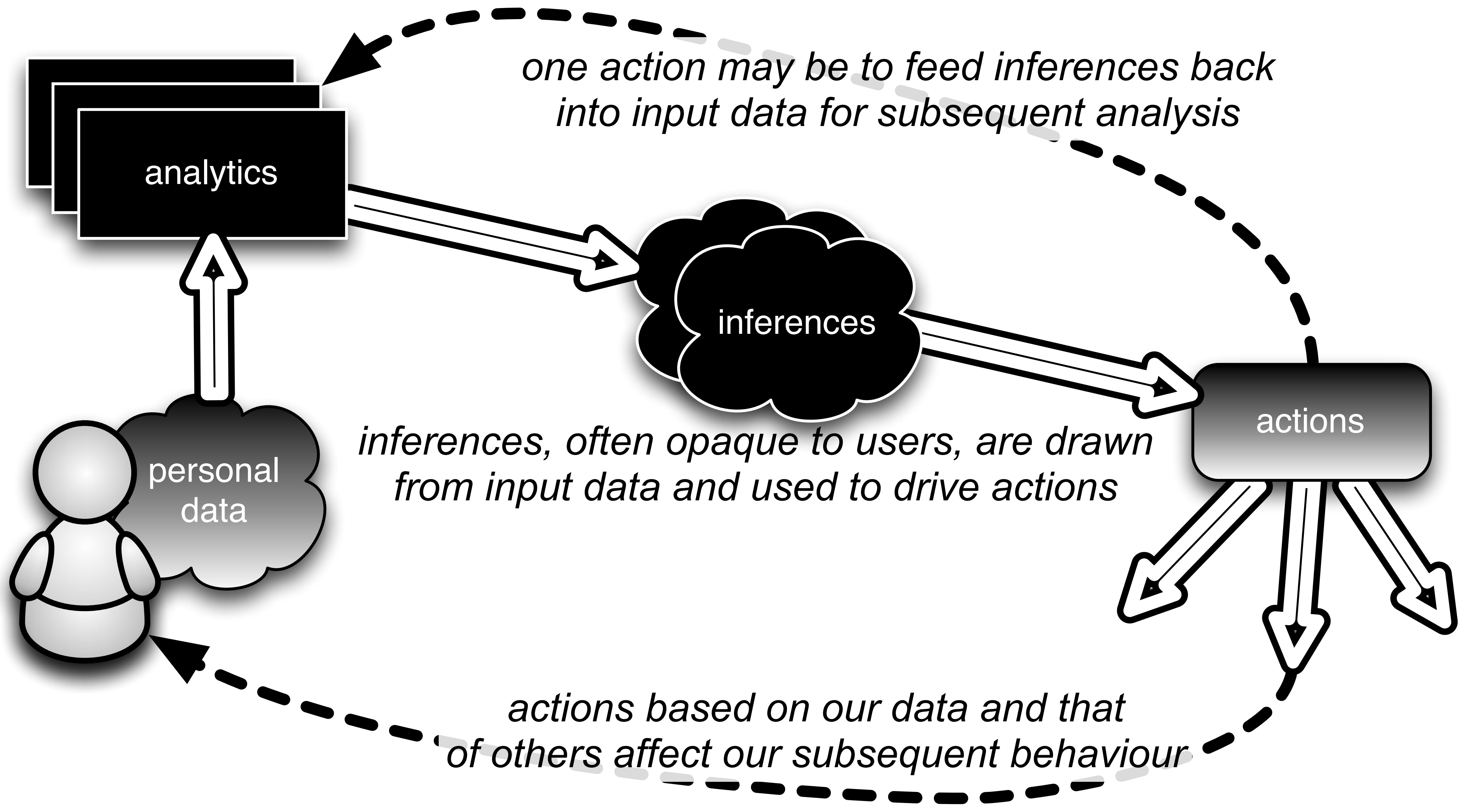
Human-Data Interaction

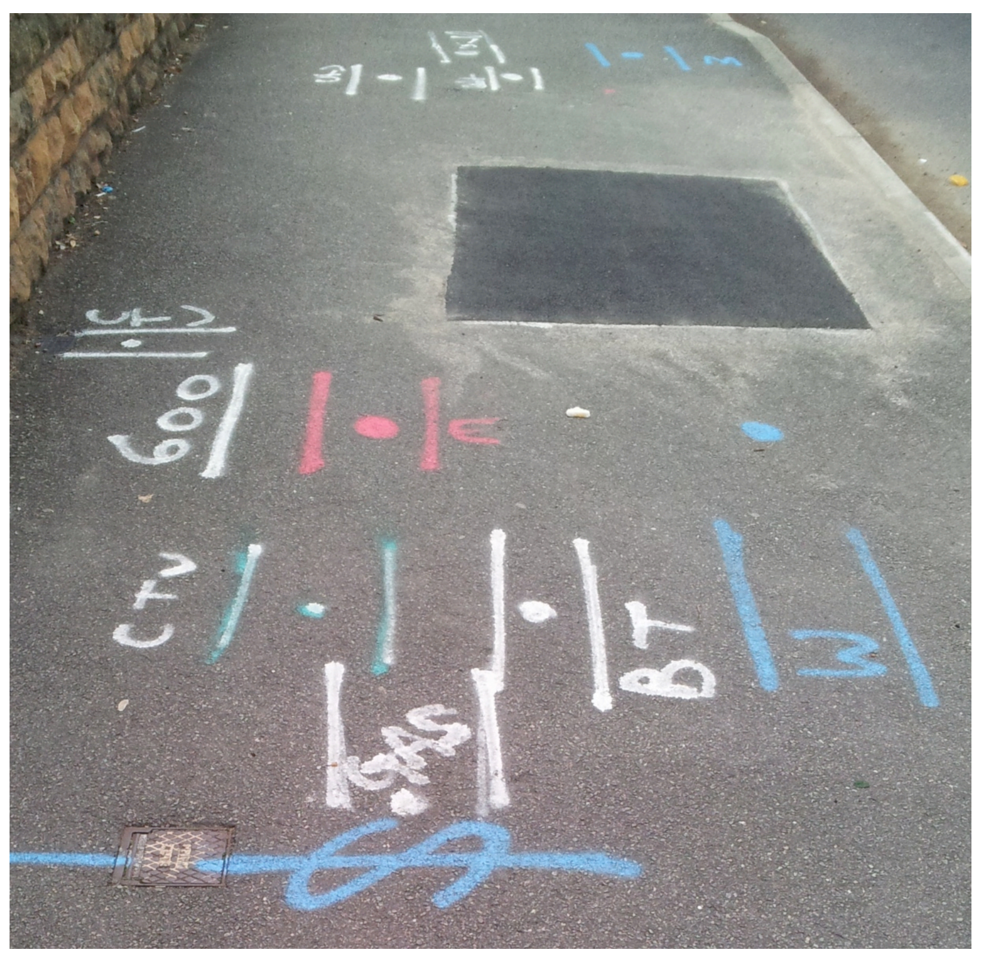
UNIVERSITY OF CAMBRIDGE

The trend towards truly ubiquitous and pervasive computing raises the issue of how people should interact with **data**

Three key challenges: providing **legibility**; enabling **agency**; supporting **negotiability**

Current deployment of home network infrastructure


analytics
inferences
actions
personal data
one action may be to feed inferences back into input data for subsequent analysis
inferences, often opaque to users, are drawn from input data and used to drive actions
actions based on our data and that of others affect our subsequent behaviour



The occasional visibility of physical infrastructures

We need ***User-Centric Systems***, to bring infrastructures under user control while still enabling them to exist for the benefit of all

Richard.Mortier@cl.cam.ac.uk