Bridging the Security Gap with Decentralized Information Flow Control

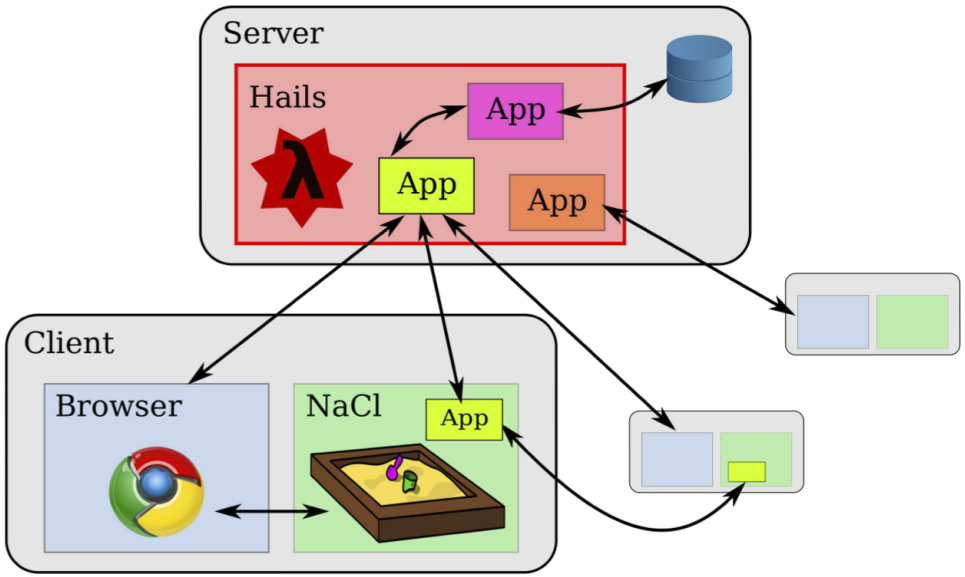
Adam Belay, Andrea Bittau, Dan Boneh, Pablo Buiras*, Daniel Giffin, **Amit Levy**, Ali Mashtizadeh, David Mazieres, John Mitchell, Alejandro Russo*, Amy Shen, Deian Stefan, David Terei, Edward Yang, Nickolai Zeldovich†

Stanford, †MIT, *Chalmers

Project Goal

Make it possible for programmers who are not security experts to build secure web applications

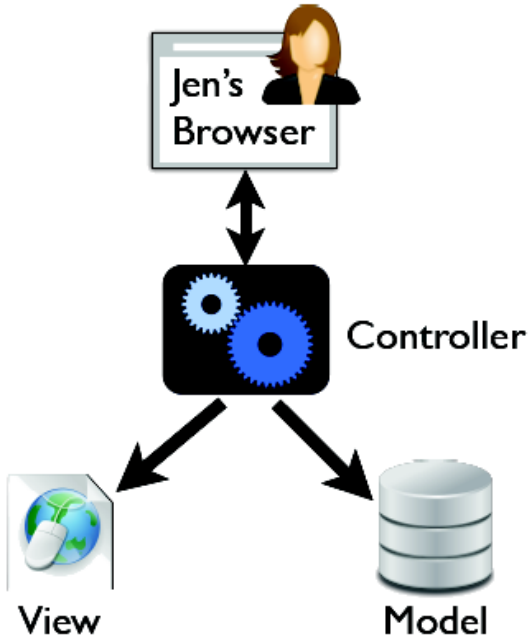
Architecture



- **Server**: Language framework for building secure apps
  - Hails, LIO
- **Client**: Sandboxing and browser security
  - Starlight, Dune

The Server Side Today: Web Apps

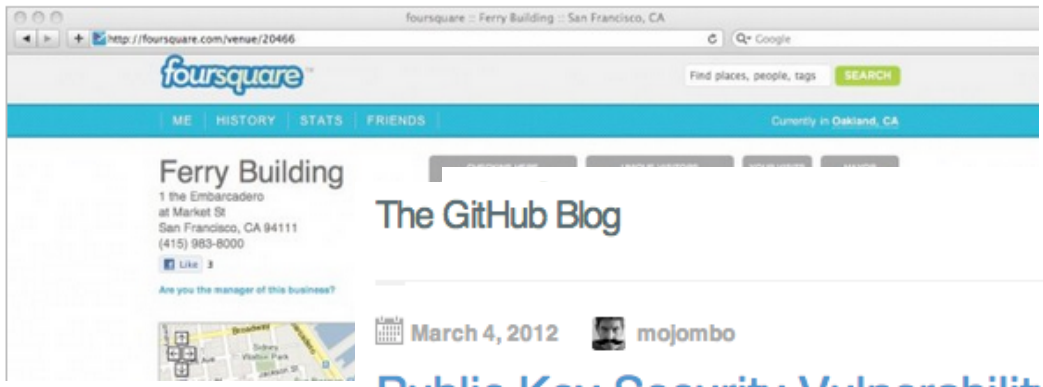
- Most apps structured around MVC (Model-View-Controller)
  - Rails, Django, Struts, .NET, others…
- Useful for compartmentalizing development



Why is the Web so &$@*ing Broken?!

Foursquare vulnerability exploited: 'private' location data captured

Jun. 30, 2010 (10:15 am) **By:** *Andy Carvell*



The GitHub Blog

March 4, 2012 mojombo

Public Key Security Vulnerability and Mitigation

Confluence Security Advisory 2012-05-17

Added by Andrew Lui [Atlassian Technical Writer], last edited by Sarah Maddox (Administrative Account) on Aug 08, 2012

This advisory discloses a **critical** security vulnerability that exists in all versions of Confluence up to and includin

- **Customers who have downloaded and installed Confluence** should upgrade their existing Confluence i
- **Enterprise Hosted customers** need to request an upgrade by raising a support request at http://support.a project.
- **JIRA Studio and Atlassian OnDemand customers** are not affected by any of the issues described in this

The Server Side Today: Web Apps

- No notion of security policies
- Ad-hoc security checks throughout applications
  - Easy to forget a check (e.g. GitHub mass assignment volnerability)
  - Extracting the policy requires looking at the **whole application**
  - Often breaking MVC abstraction

Minding the Gap

Lots of research on secure systems

Jif, HiStar, Asbestos, Nexus, Ur/Web, ...

Technologies not adopted

- Modify entire stack
- Not approriate for dynamic systems like the web
- Policies are hard to write
- No guide for structuring applications

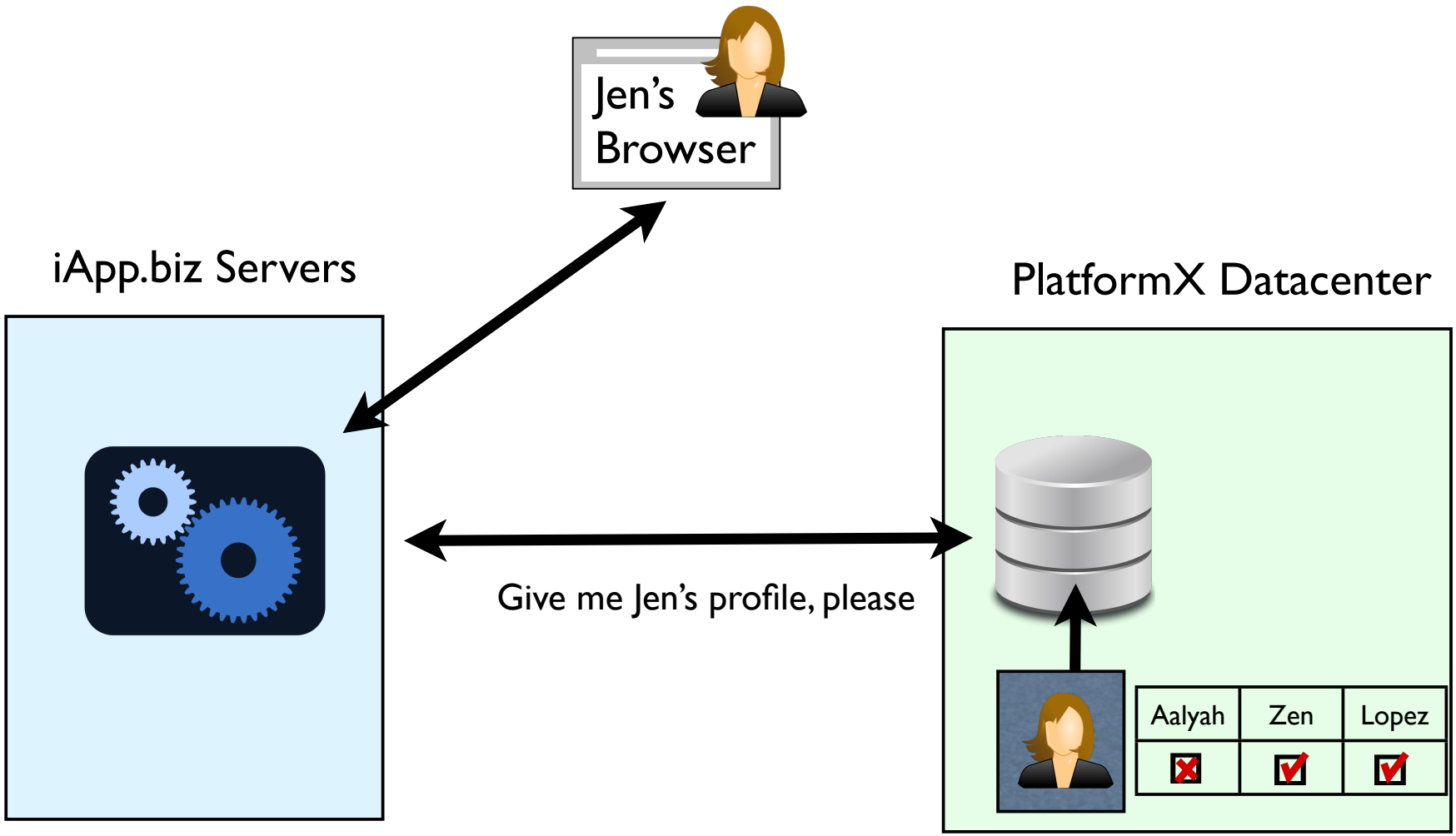
Hails: A web platform framework

- Hails targets web *platforms*, not just apps
  - All code is untrusted and potentially malicious

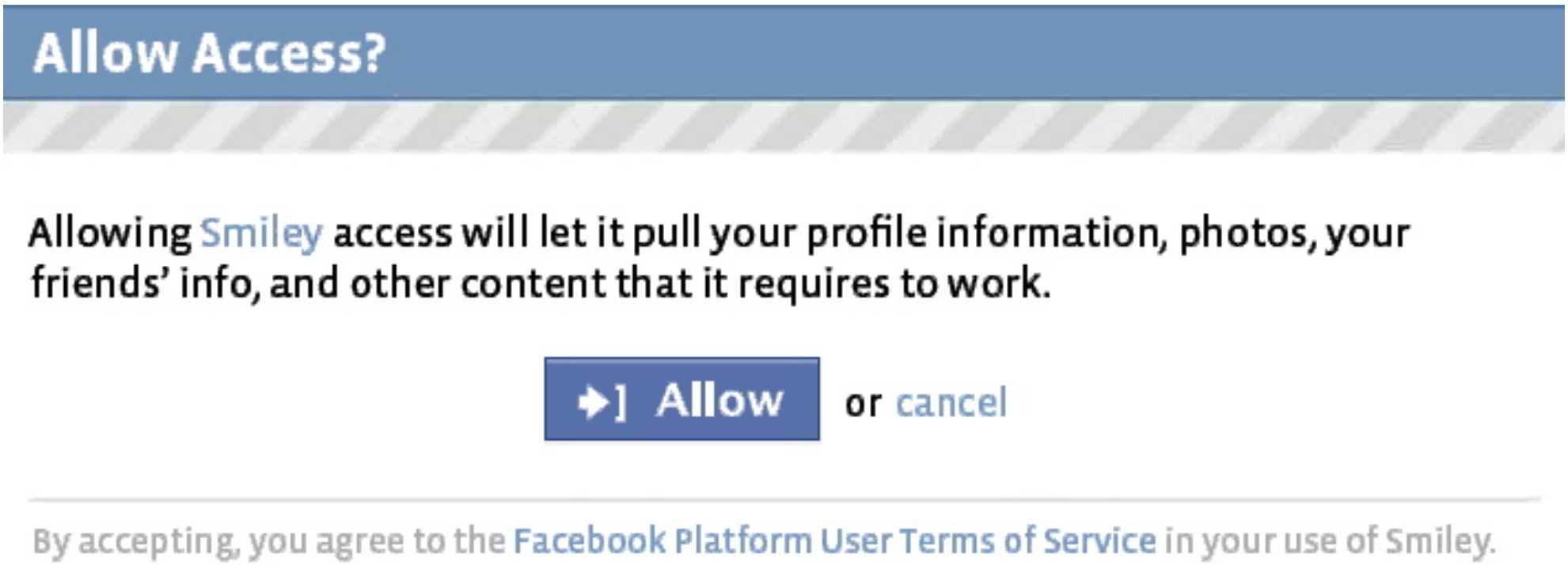
Goals

- Suitable for web *platforms*
- Usable by web developers
  - Easy to write policies
  - Easy to write the rest of the app
- Deployable today
  - Change as little of the stack as possible

The Server Side Today: Web Platforms



Current Solution



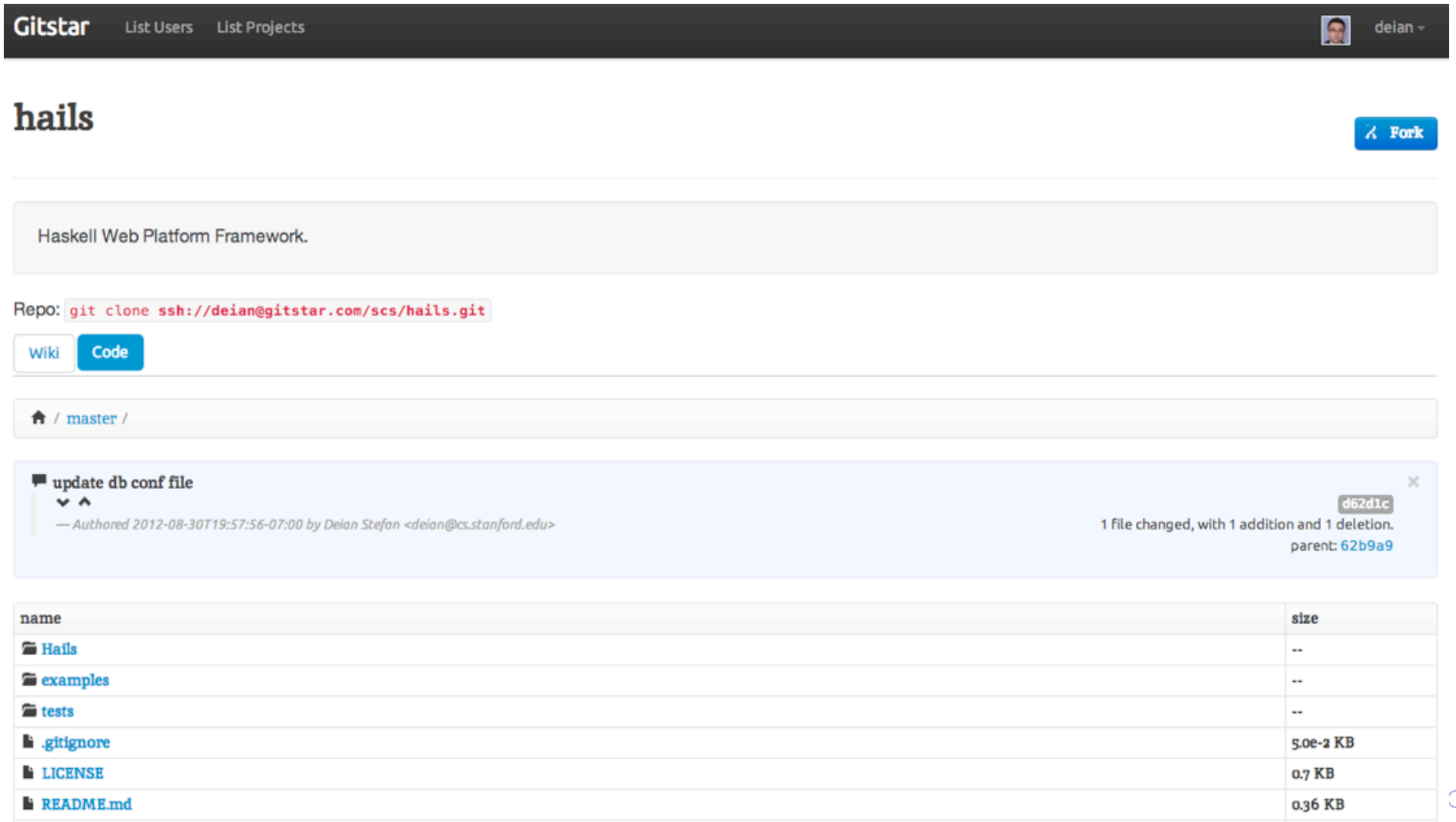
Change the hosting model

- Developers host apps on in their own datacenters
- Platforms enforce security contractually (e.g. terms of service)

Hails: A new approach

- *Platforms* host apps on their own hardware, on top of Hails
- Use information flow control to **ensures** apps obey security policies

Case Study: Gitstar.com



Adding Policy to MVC

- New paradigm: Model-**Policy**-View-Controller
  - Policy specified independantly
  - No policy in the Model, View or Controller
- Hails has two types of third-party code
  - Model-Policies (MPs)
  - Provide data model and policy
  - View-Controllers (VCs)
  - Web server executables that link to MPs

Trust Model in Hails

- View-Controllers are completely untrusted
  - Includes most of the interesting functionality, like UI
- Model-Policies must only be trusted with the data they define
  - Users have to trust that they set good policies.
- Hails uses information flow control (IFC) do enforce policies on data models, end-to-end

MPs and VCs in Gitstar

- The Gitstar platform provides:
  - MPs for projects and users
  - A VC for managing projects and users (http://www.gitstar.com)
- Third-party authors:
  - Source code browser
  - Wiki
  - Follower app
  - Their own MPs
- In fact, nothing special about the Gitstar VC

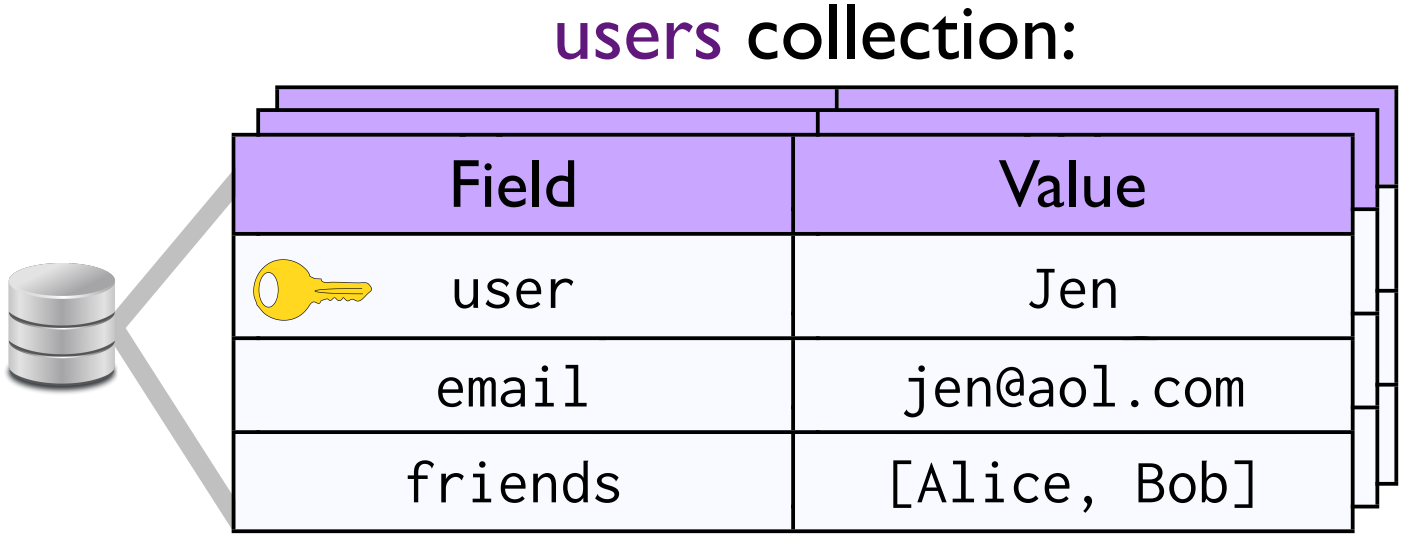
MPs and VCs

A closer look…

Model Policy

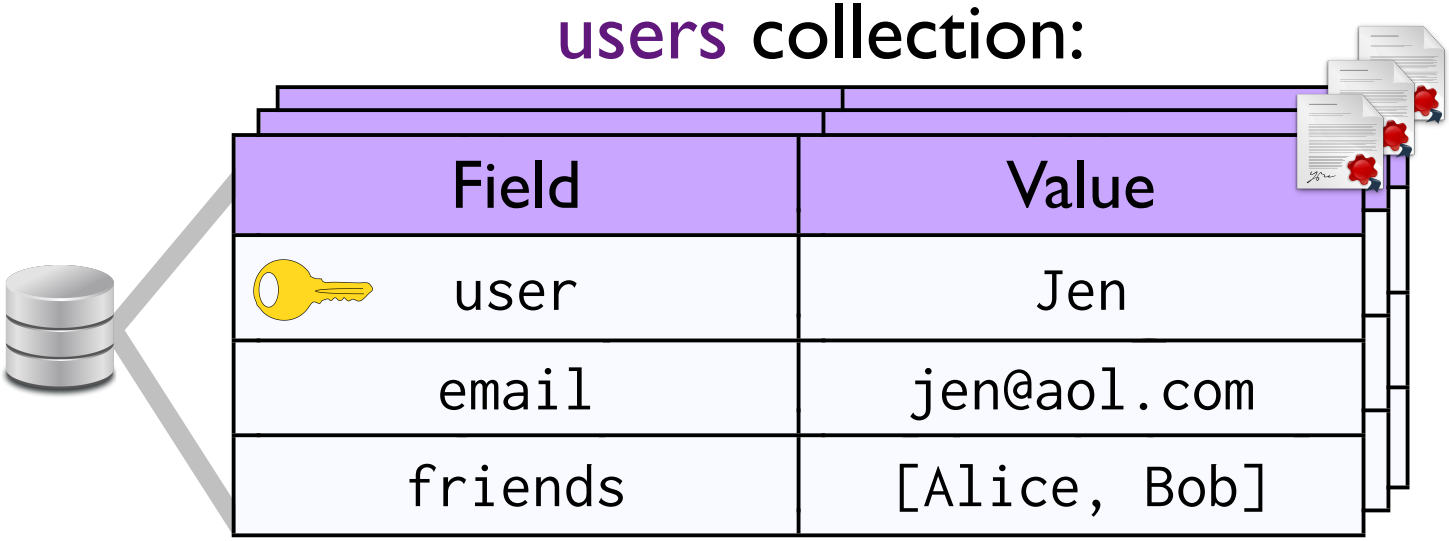
A document oriented data-store:

- Documents are stored in collections, stored in databases
- Semi-structured schema with flexible data-types



Model Policy

- Web app data *already* encodes policy
- Function from a document to a policy



```
collection "users" $ do
    access $ do
      readers ==> anybody
      writers ==> anybody
    field "user" key
    document $ \doc -> do
      readers ==> anybody
      writers ==> ("user" „from„ doc)
    field "email" $ labeled $ \doc -> do
      readers ==> ("user" „from„ doc)
              \/ fromList ("friends" „from„ doc)
      writers ==> anybody
```

View Controller

- A VC is a web request handler
- Implement UI and external API
  - Source code viewer, RSS feed, Wiki editor,...
- Handle all data persistence through MPs
- Low barrier, since new VCs can reuse existing MPs

Bugs in VCs are manifested as broken features – never as vulnerabilities


www.gitstar.com/scs/hails
Gitstar
List Users
List Projects
deian
hails
Haskell Web Platform Framework.
Repo: git clone ssh://deian@gitstar.com/scs/hails.git
Wiki
Code
Code viewer VC
/ master / examples / SimpleParams.hs
{-# LANGUAGE OverloadedStrings #-}
module SimpleParams (server) where
import qualified Data.ByteString.Lazy.Char8 as L8
import LIO
import LIO.DCLabel
import Hails.HttpServer
import Hails.Data.Hson
server :: Application
server _ lreq = do
req <- unlabel lreq
let ldoc = labeledRequestToLabeledDocument lreq
doc <- unlabel ldoc
return $ case pathInfo req of
("login":_) -> Response temporaryRedirect307
[("x-hails-login",""),(hLocation,"/")] ""
_ -> Response ok200 [] $ topHtml (labelOf lreq, req) (labelOf ldoc, doc)
www.gitstar.com/scs/hails
Gitstar
List Users
List Projects
hails
Haskell Web Platform Framework.
Repo: git clone ssh://deian@gitstar.com/scs/hails.git
Wiki
Code
Wiki VC
Home
Pages
Hails: Protecting Data Privacy in Untrusted
Hails is a platform and web framework that leverages Information Flow Control (IFC) to suppo
applications interacting and processing private data.
Resources
Brief motivation and architecture overview
Tutorial (slightly outdated)
Installation
You can compile and install Hails as usual with cabal-dev :
$ cabal-dev instal-deps
$ cabal-dev install
Launching an app
If you define your main application (named server) in YourAppModule.hs , you can launch it
GitStar Project MP
+

Goals

- Suitable for web *platforms*
- Usable by web developers
- Deployable today

Evaluation: Usablility

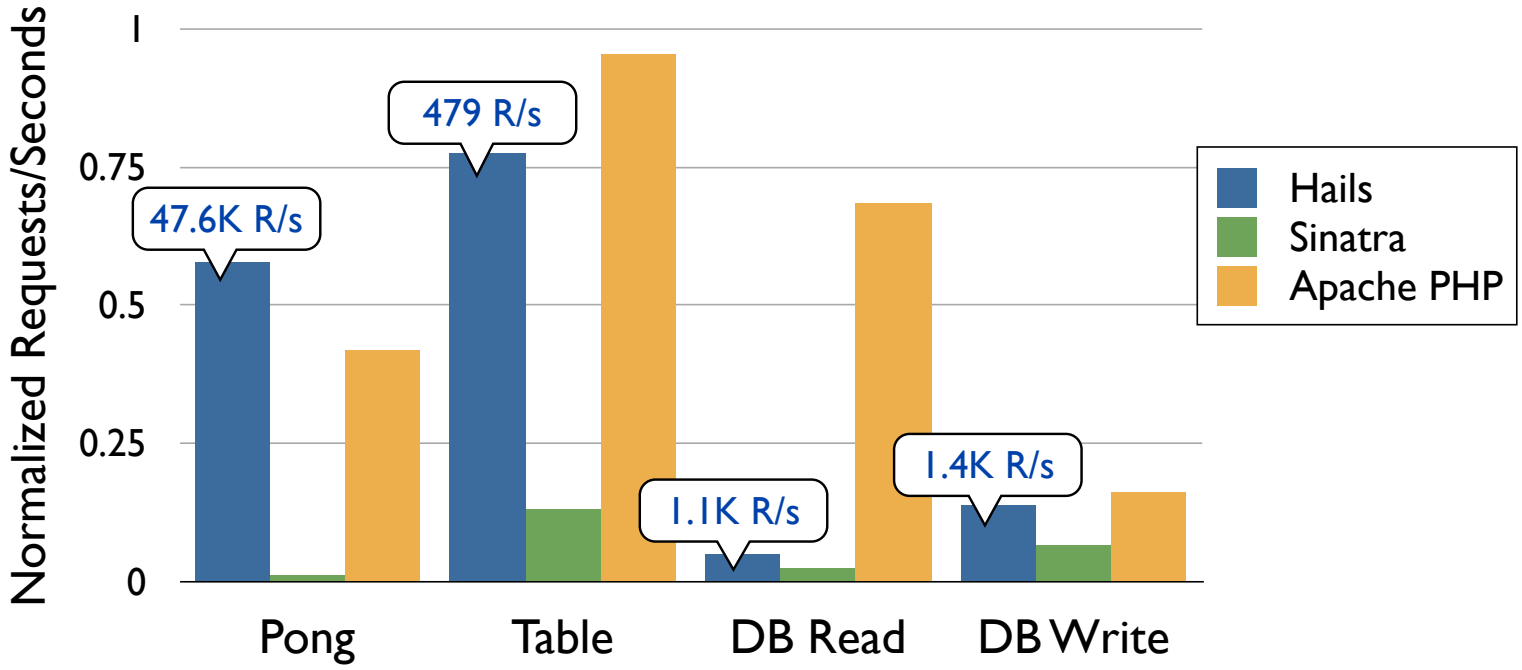
$\surd$ MPVC simplified reasoning about security $\surd$ Hails rendered common security bugs futile

χ Need scaffolding tools

χ ~~Writing policies is hard~~.

? Better with new policy DSL

Evalutaion: Performance



Limitations / Present & Future Work

- Confined to Haskell
  - Now - cjail
  - Future - Dune
- Covert channels
  - Internal timing closed ([ICFP 2012])
  - External timining - mitigation
  - How much to mitigate?
  - More work to do...
  - Cache-based timing attack

tl;dr

- Current platforms: functionality vs. privacy
- Hails platforms guarantee security end-to-end
  - Host apps on platform
  - Make policy explicit
  - Enforce policy with information flow control

```
$ cabal install hails
```

`http://gitstar.com` `http://hails.io/`