David C. Knapp

david.c.knapp@gmail.com
(650) 720-5831
http://careers.stackoverflow.com/dknapp
https://github.com/Falmarri

Objective

To spend every day with talented teammates making things people love that solve interesting problems.

Work Experience

Software Infrastructure Engineer — August 2012 - Present

iPlant Collaborative at the University of Arizona, Tucson, AZ

iPlant is a National Science Foundation open source initiative intended to create an innovative cyberinfrastructure designed to run inter-operable scientific analysis software and big data sharing services to solve critical challenges in the life sciences.

- **Designed user and group management system with Pyramid web framework and Neo4j.**
  System tied LDAP management, iRODS user accounts and quota allocations, group memberships, inter-application communication and centralized application registry.

- **Built OAuth2 service provider using JSON Web Tokens.**
  Implemented draft-ietf-oauth-jwt-bearer-07 OAuth2 workflow using the group management system for application authorization and scope granting procedures. Implemented references clients in python and scala.

- **Developed tools for data analysis.**
  With 15,000+ users in our system running analyses on hundreds of terabytes of data, the integrated user management system made tracking usage of scientific tools and data possible. Data is queried using a combination of Neo4j's cypher language, iRODS searches, and OAuth token usage data.

Software Engineer — August 2010 - August 2012

Ephibian, Tucson, AZ

Ephibian is the main engineering contractor for designing and building the Blink electric vehicle charging hardware platform, back end data infrastructure, and native Android and iPhone applications.

- **Wrote driver to control a J1772 interface as a native C Python module.**
  Using native code freed the global interpreter lock while communicating with the J1772 via GPIO pins.

- **Built driver to read from the charger's energy meter as a native C python module.**
  Since reads from the meter over RS-485 could take upwards of 4 seconds with several failures, using native C code to prevent the global interpreter lock from forcing context switches while communicating

- **Built and managed the network infrastructure and server software that all the Blink chargers connected to.**
  Used postgreSQL partition tables to handle 50,000 rows of charge data and associated metadata per day. Server could automatically deliver over-the-air updates to units, including kernel updates, with the ability to rollback on failures.

- **Developed Android app for users to view charging information.**
  App allowed users to view their current charging status, be notified of errors that might occur while charging, as well as stop or restart a charging session. App also displayed a map of all charger locations using Google Map APIs so users could search for a charger near them.

Education

The University of Arizona	**May 2010**
B.S., Electrical Engineering	*Minors: Math, Computer Engineering*