Open Source, Mobile, Cloud, and Medical Images

Michael J Pan, CEO
nephosity

Synopsis

- What are we building?
- What problem are we trying to solve?
- Demo
- Why are we using open source?
- Which open source packages do we use?

What are we building?

a mobile collaboration ecosystem

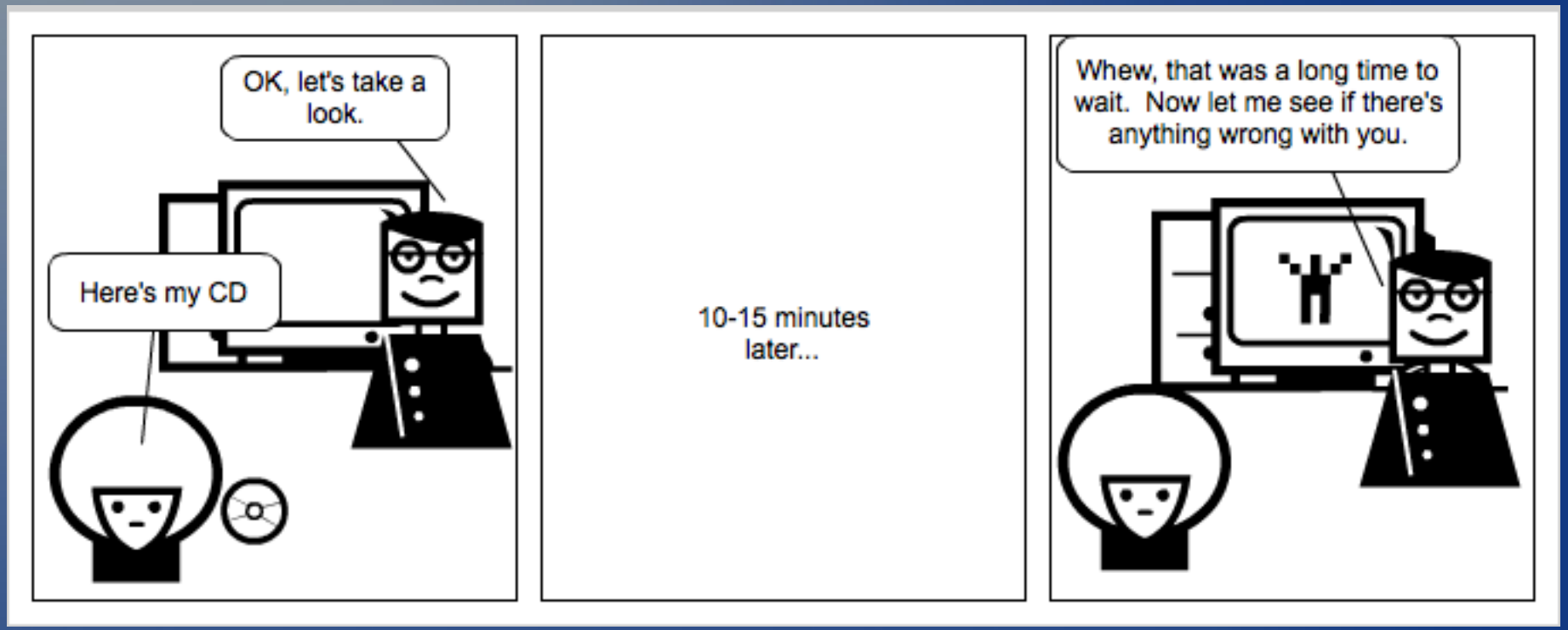
for

medical images

The problem

Current workflows
involving medical images are
cumbersome and non-collaborative

Example 1

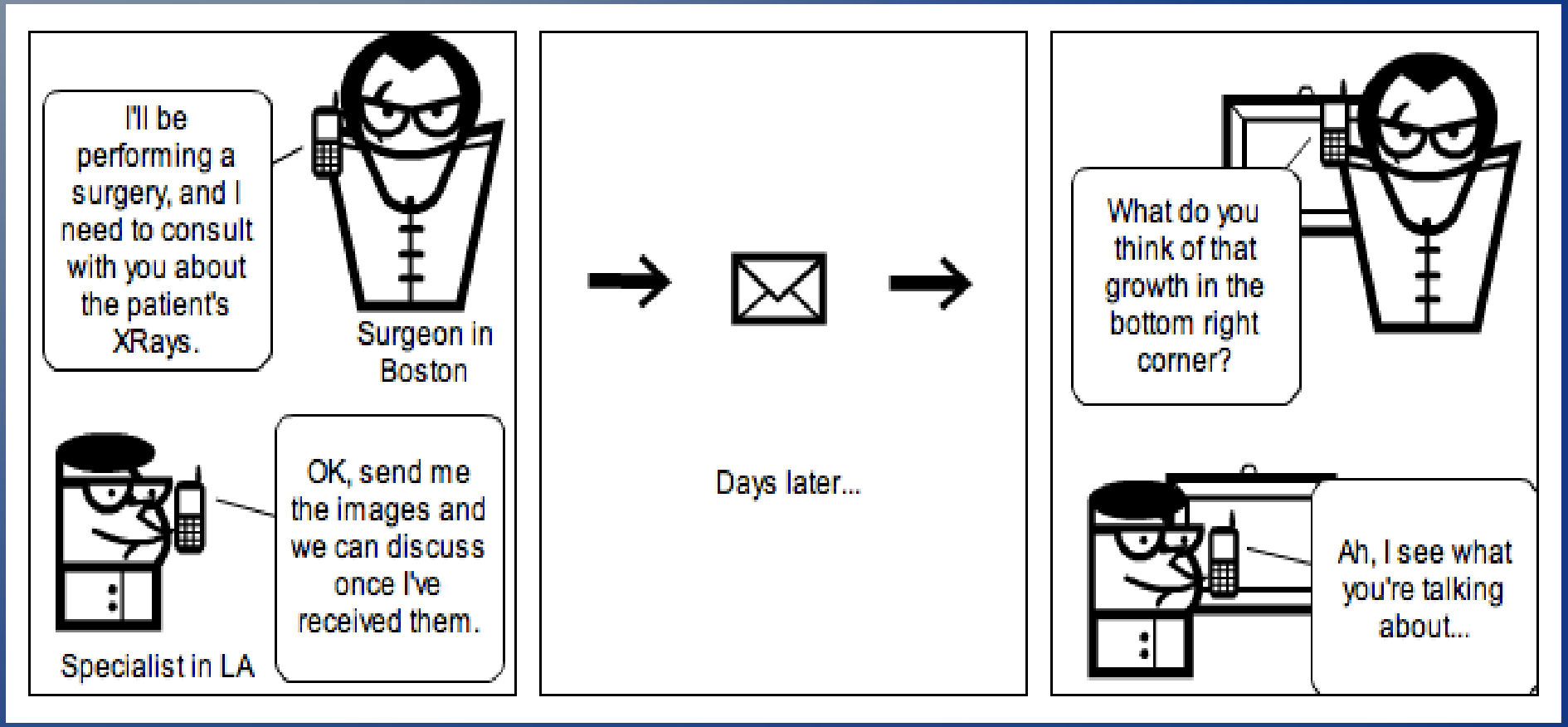

OK, let's take a look.
Here's my CD
10-15 minutes later...
Whew, that was a long time to wait. Now let me see if there's anything wrong with you.

Example 1

- You get an MRI (or an Xray) at an imaging center
- The imaging center gives you a CD
- You bring the CD to your doctor
- Your doctor loads the CD on his desktop (10-15min)
- Your doctor views the images and gives you a diagnosis

Problem: It takes a long time to load images. A doctor does this 20-30 times a day

Example 2


I'll be performing a surgery, and I need to consult with you about the patient's XRays.
Surgeon in Boston
OK, send me the images and we can discuss once I've received them.
Specialist in LA
Days later...
What do you think of that growth in the bottom right corner?
Ah, I see what you're talking about...

Example 2

- Your doctor wants to consult a doctor in another hospital
- Your doctor orders a CD of your MRI and sends that CD to the other doctor
- The other doctor receives and loads the CD
- The 2 doctors discuss on the phone

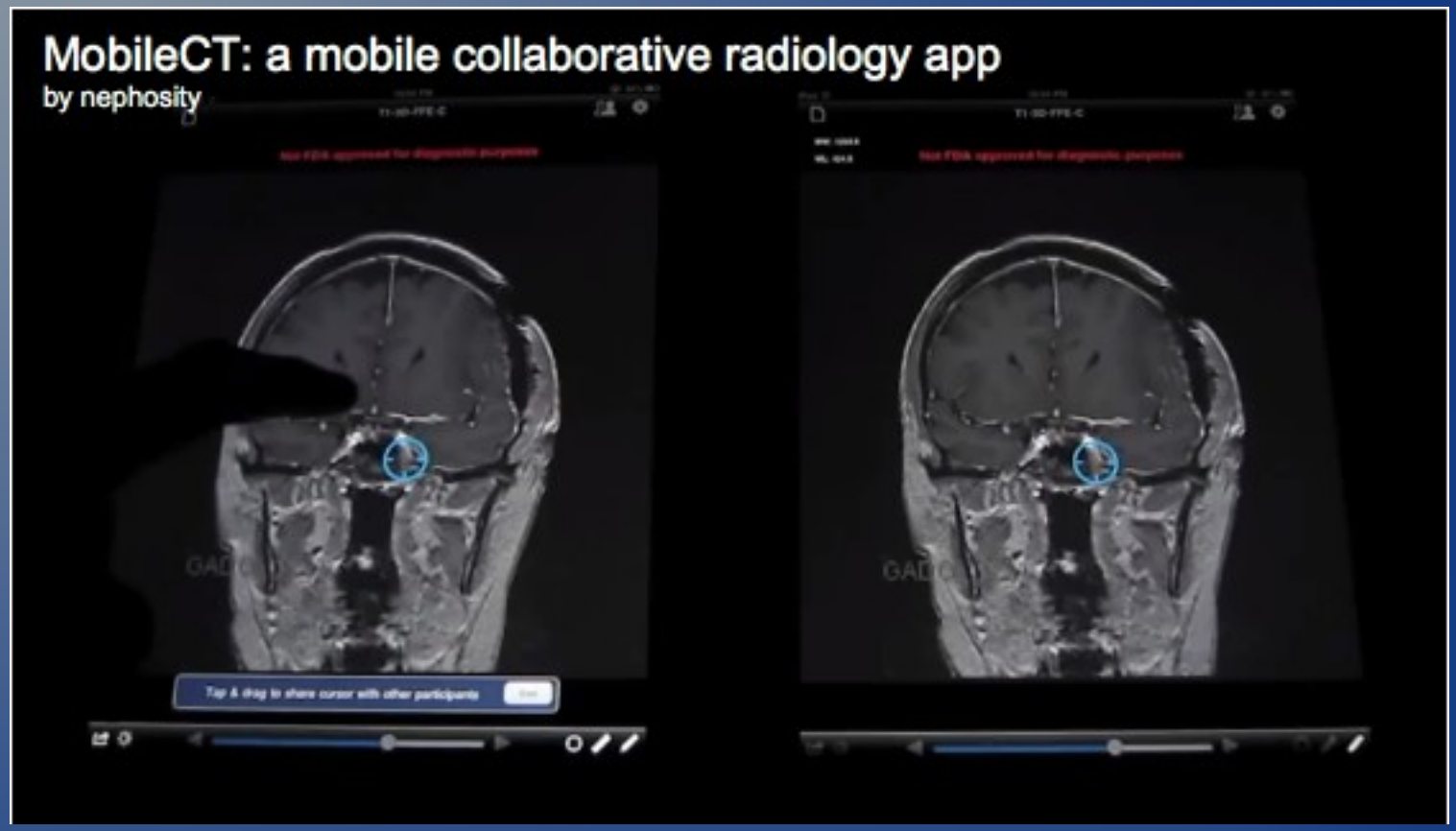
Problem: creating and sending CDs require time, and no shared view during collaboration

Our solution

Build a mobile collaboration ecosystem to eliminate friction in viewing and collaborating over medical images. 2 components:

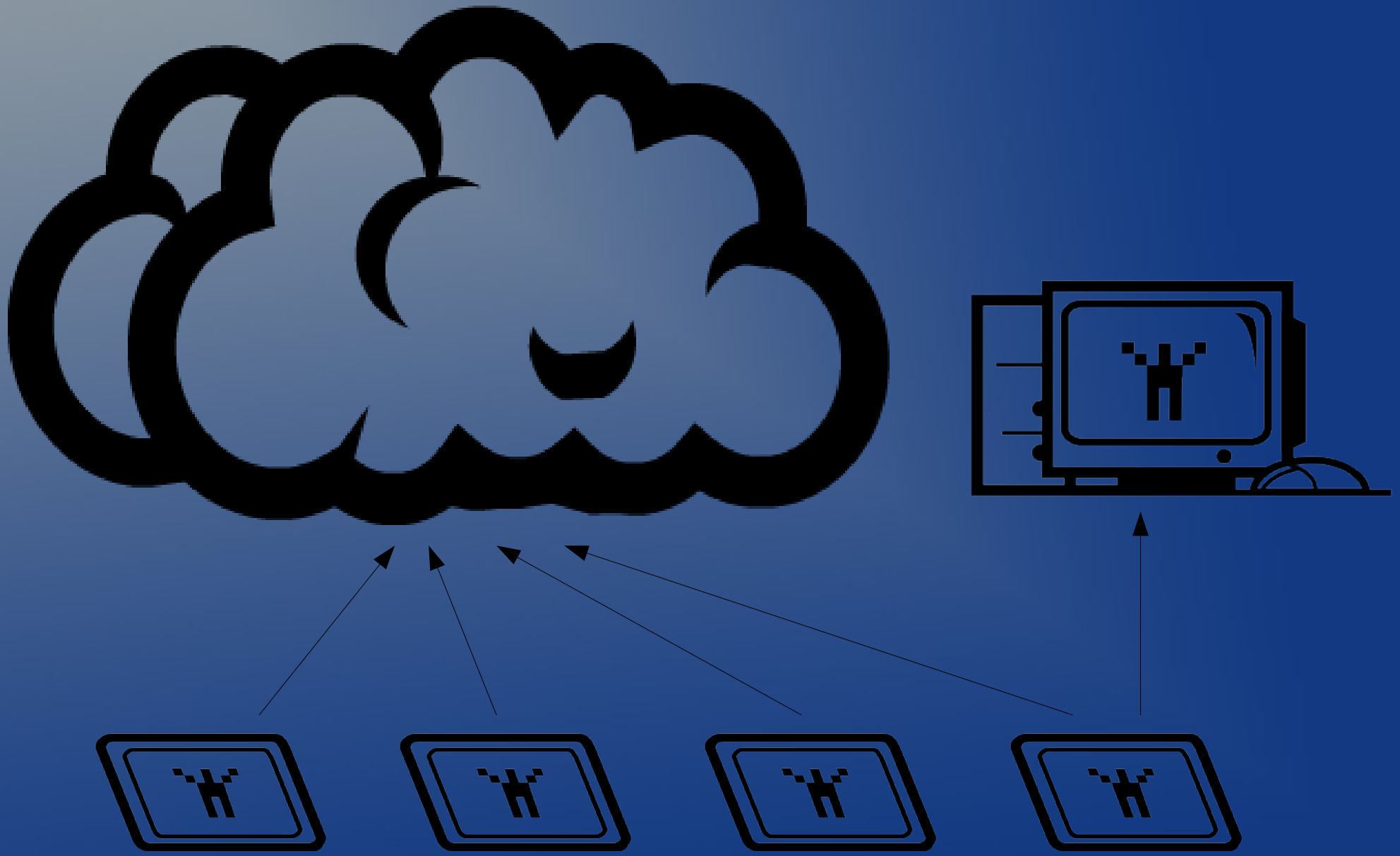
- nephosCT is the cloud server backend that supports collaboration and delivers medical images in real time
- mobileCT is the mobile collaborative teleradiology app for viewing and collaborating on your iPad

Product demo video



Click to play in browser

System design overview



Why use open source?

We built this system using lots of open source packages

because...

We don't know that much

(you understand the problem
better than we do)

We're not that smart

(you can come up with
better solutions than we can)

We're kinda lazy

(why do something if you have already done it for us?)

and...

we're busy /
have too much to do

(angry birds is very time consuming ;)

How do we use open source?

- In the product
  - Desktop server
  - Cloud server
  - Mobile app
- Productivity
  - Developer tools
  - Project management software
  - Quality assurance
  - Packaging tools

and now...

An unordered, non-comprehensive list of open source packages that we use

GUI

- QT / PySide
- three20

QT / PySide

- QT is C++ GUI library
- PySide is Python bindings for QT
- Developed by Nokia
- LGPL license
- Build cross-platform GUI for our desktop server

three20

- An UI library for the iOS devices
- Originally developed by Facebook
- Apache license
- Build GUI for our iPad app

Cloud service

- Eucalyptus
- Mongrel2
- Celery

Eucalyptus

- An IaaS software, open source equivalent of Amazon's EC2
- Developed by Eucalyptus Systems
- GPL license
- Deploy private clouds

mongrel2

- Language agnostic web server
- Developed by Zed Shaw
- BSD license
- Our web server / gatekeeper to our cloud

celery

- Distributed task queuing system
- Originally developed by Django
- BSD license
- Pool and manage tasks to process medical images uploaded to our cloud

Messaging

- Rabbitmq
- 0mq
- UDT
- Zeroconf

rabbitmq

- Brokered messaging with persistence
- Developed by VMWare / SpringSource
- Mozilla Public license
- AMQP messaging backend for celery

0MQ

- Super fast, brokerless messaging library
- Developed by iMatix
- LGPL
- Messaging backend for mobile, mongrel2, various backend cloud components

UDT

- UDP based Data Transfer
- Developed by Yunhong Gu (main developer of Sector/Sphere)
- BSD license
- Used by SectorFS for high bandwidth transfer

Zeroconf

- Bonjour
  - Apple's implementation of the zeroconf spec, for Mac OSX and Windows
  - Apple Public source license
- Avahi is the open source implementation of zeroconf for other *nix and BSD
  - LGPL
- PyBonjour is the Python wrapper to a system's underlying zeroconf library
  - MIT license
- Save users from needing to configure the connection to their desktop server / internal cloud

DICOM

DICOM (Digital Imaging and Communications in Medicine) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol.

- GDCM
- dcm4che

GDCM

- Library for reading / writing DICOM images. Currently no network capabilities (handles only file format portion of DICOM standard)
- Developed by Mathieu Malaterre
- BSD license
- Library to read medical images

dcm4che

- An open source clinical image and object management platform
- dcm4che.org
- MPL/GPL/LGPL
- DICOM network library to query/download DICOM files from hospital PACS (Picture archive & communication system)

Data storage / distribution

- SectorFS
- MongoDB
- SQLite

SectorFS

- Super fast fault-tolerant distributed file system
- Developed by the National Center for Data Mining (NCDM) at the University of Illinois at Chicago (UIC)
- Apache 2.0 license
- Distributing/replicating large files (ie the DICOM images) across our cloud

MongoDB

- Distributed NoSQL database
- Developed by 10gen
- AGPL license for database, Apache license for drivers
- Replicating image metadata

SQLite

- Embedded SQL database
- Developed by Hwaci
- Public domain
- Configuration / metadata caching
- Also used by mongrel2 for configuration

Other packages in our code

- SBJson (ObjC JSON library)
- mext/Reaction (event processing in Python)
- chardet (language detection)
- threadpool

Open source productivity software

- Subversion (version control)
- nose (unittesting in Python)
- coverage (test coverage in Python)
- trac (project management)
- pyinstaller (multi-platform packaging of our desktop server)
- InnoSetup (installer package creator of our desktop Windows server)

Our open source contributions

- pomsets (workflow management for your cloud)
- cloudpool (execution pool in the cloud)
- pypatterns (common patterns in Python)
- currypy (de/serializable curried objects)
- Patches to open source projects that we use

You ask: Any other open source?

- Lots!
- Continue to investigate new technologies, new open source packages
- Many are still in evaluation, so not listed here

Summary

- We are building mobile collaboration for medical images
- Touches the latest, hottest buzzwords-- mobile, collaboration, medical, cloud
- We are doing it using lots of open source, plus a bit of our “secret sauce”
- We contribute some of our secret sauce back as open source

Have medical images?

- Buy our app!
- Cloud beta Q1 2012

Available on the iPad
App Store

- http://nephosCT.com

Questions?

nephosity
mobilize the cloud

Michael J Pan, CEO
mjpan@nephosity.com