

Place in Time: Contextualizing Crises

John Kelley, Sophia Liu, Matt Novinger and Lon Riesberg

November 10, 2008

Web Services and Web 2.0 – Professor Ken Anderson

Crises

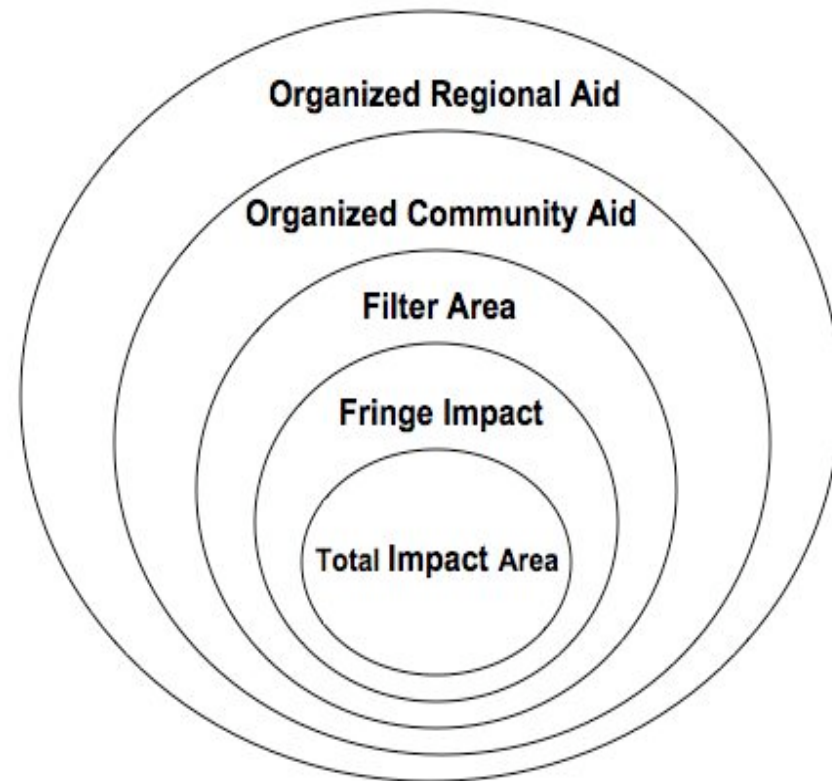
- May be depressing or morbid...but
- Real-life events that draw attention to serious vulnerability issues that face our society
- Crises are increasing due to our vulnerabilities
- Natural, technological, human-induced hazards; climate change, wars, crime, political injustice

What I Proposed

- Visual Narrative Maker
- Visual Annotation Tool
- Spatial Guide Maker
- Map Timeline Mashup

Temporal and Spatial

Stage 0: PRE-DISASTER State of social system preceding point of impact
Stage 1: WARNING Precautionary activity includes consultation with members of own social network
Stage 2: THREAT Perception of change of conditions that prompts survival action
Stage 3: IMPACT Stage of "holding on" where recognition shifts from individual to community affect and involvement
Stage 4: INVENTORY Individual takes stock, and begins to move into a collective inventory of what happened
Stage 5: RESCUE Spontaneous, local, unorganized extrication and first aid; some preventive measures
Stage 6: REMEDY Organized and professional relief arrive; medical care, preventive and security measures present
Stage 7: RECOVERY Individual rehabilitation and readjustment; community restoration of property; organizational preventative measures against recurrence; community evaluation



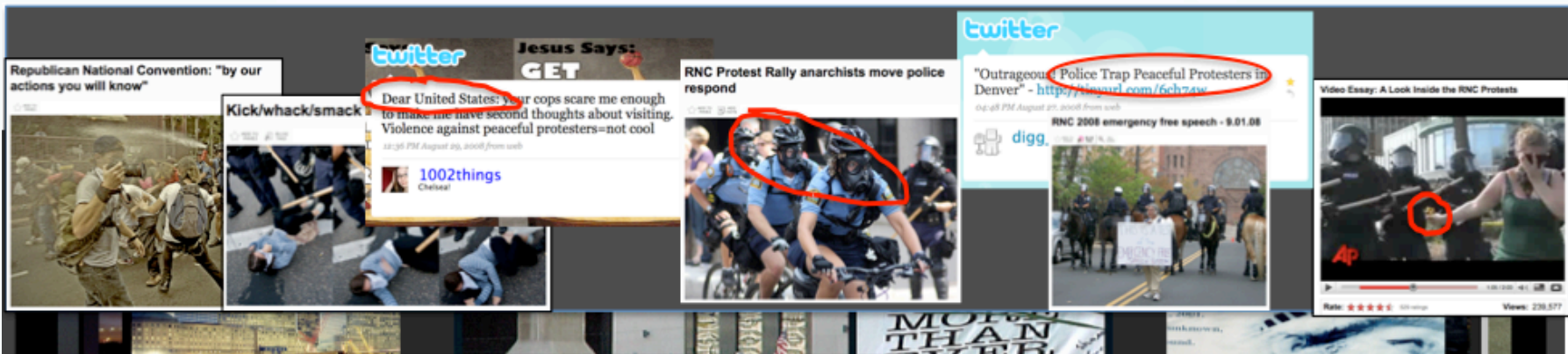
Dynes, R. R. 1970. *Organized Behavior in Disaster*. Heath.

Powell, J. 1954. *An Introduction to the Natural History of Disaster*. Univ. of MD: Disaster Research Project.

Place in Time Mashup

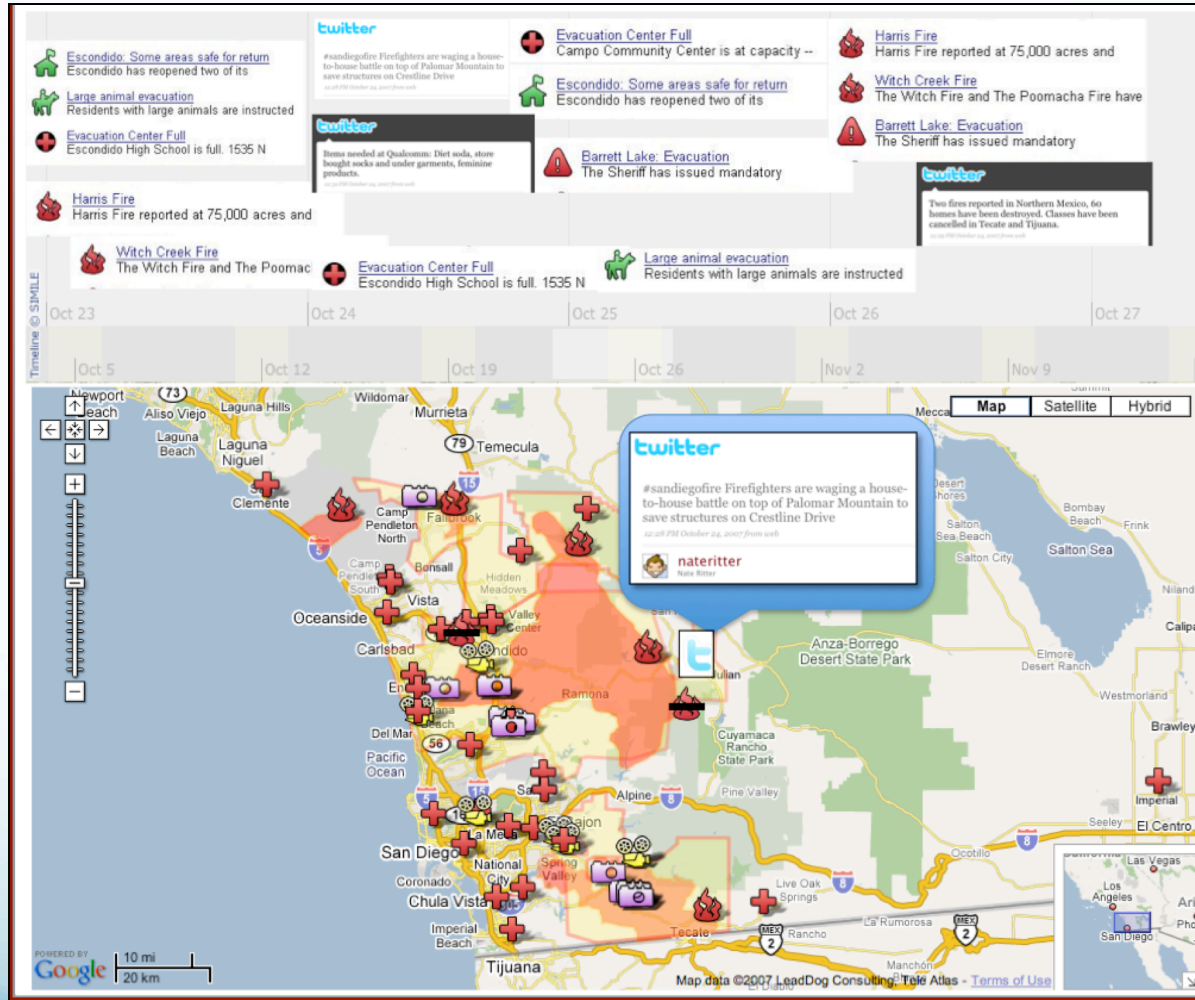
- View photos about a crisis by its spatial and temporal context
- Take photos from a mobile device, upload it to Flickr and post it in the mashup
- Filter the data shown in mashup based on text and location

First Vision



Title: Aggressive Riot Police
Theme: civil liberties
Tags: RNC, DNC, protest, tear gas
Description: This is a story about how the riot police have become extremely aggressive in dealing with peaceful protesters using their right to free speech in public places. As a nation, we should be concerned about how our civil liberties are at stake.

Second Vision



Final Prototype

The screenshot displays a web application interface for tracking events. At the top, a horizontal timeline spans from March 13 to March 21. A vertical blue bar indicates the current time is approximately 10:00 AM on March 17. Below the timeline is a map of the Pittsburgh area, showing various locations and roads. Several colored pins are placed on the map, corresponding to the legend. A white speech bubble with a blue border is positioned over a red pin near Verona, containing the text "#accident wow I see a huge green cloud!".

Legend:

- #hazmat a big hazmat truck just tipped
- OMG #hazmat truck just ran off the road
- #hazmat truck ran off road
- #traffic dunno what's going on
- #traffic so sick of sitting here
- #traffic #accident looks bad
- #accident wow I see a huge green cloud!

Search-:

Event Title:

Event Text:

Event Date:

Zip Code (partial works: 81 = 81000-81999):

Time After Initial Event

- 0m - 15m
- 15m - 30m
- 30m +

Map Controls: Map, Satellite, Terrain

Search Results:

Please choose a result to display-

- [1. Chemical Spill near Pittsburgh](#)

Instructions-

Scroll the timeline by dragging it left or right, then click on an icon in the timeline or map to display its contents. See the legend in the opposite corner for information on what the icon colors mean.

Front End Web Application

- Grails based application backed by a MySQL database
- Mashups Google Maps and Simile Timeline to plot events spatially and temporally
- Allows for custom icons and legends for a given group of events

Database

- Has two main tables, Episode and Event
 - The events are data points on the map/timeline and have text, titles and other information
 - The episodes serve to group the events

- Episode = 2008 Election
 - Event 1
 - Event 2
 - Event 3
 - Event 4
 - Event 5



MVC and Service Layer

- Service Layer
 - Has methods for searching the episodes and events based on text, time and zip code
 - Has a a web service call for translating lat/longs into zip codes
- One controller manages the interface.
 - Processes searches and episode requests
 - Delegates to the service for searching
 - Uses GSP pages for the views, one main page and one template (partial)

JavaScript and Ajax

- After initial page load everything is ajax.
- Passes the search parameters back and receives a template (`_search_results.gsp`)
- After an episode is chosen, event data is returned in a JSON structure along with legend and icon data

Back End REST Web Service

- JAX-RS: Java API for RESTful Web Services (JSR 311)
 - Jersey Reference Implementation
- GlassFish Application Server vs2 (Java EE 5)
 - JavaServer Pages 2.1
 - JavaServer Faces 1.2
 - Servlet 2.4
 - Enterprise JavaBeans 3.0
 - JAX-WS 2.0
 - JAXB 2.0
 - Java Persistence API, more....
- MySQL Database

Database

- Single Incident table to capture events/incidents
 - Title
 - Description
 - Location
 - Date/time
 - Related Image URL
 - Tags

REST API Support

- All CRUD at `http://{PREFIX}/webservice/resources/incidents/`
- Consume and Produce XML or JSON
- Easy to extend via JAX-RS

Mobile Component

- In 2005, there were 2 Billion mobile phones in use. In 2007, that had grown to 3 Billion. That's an active mobile device for every other person on the planet.
- Even so, Mobile Devices are largely underutilized
- HUGE opportunities

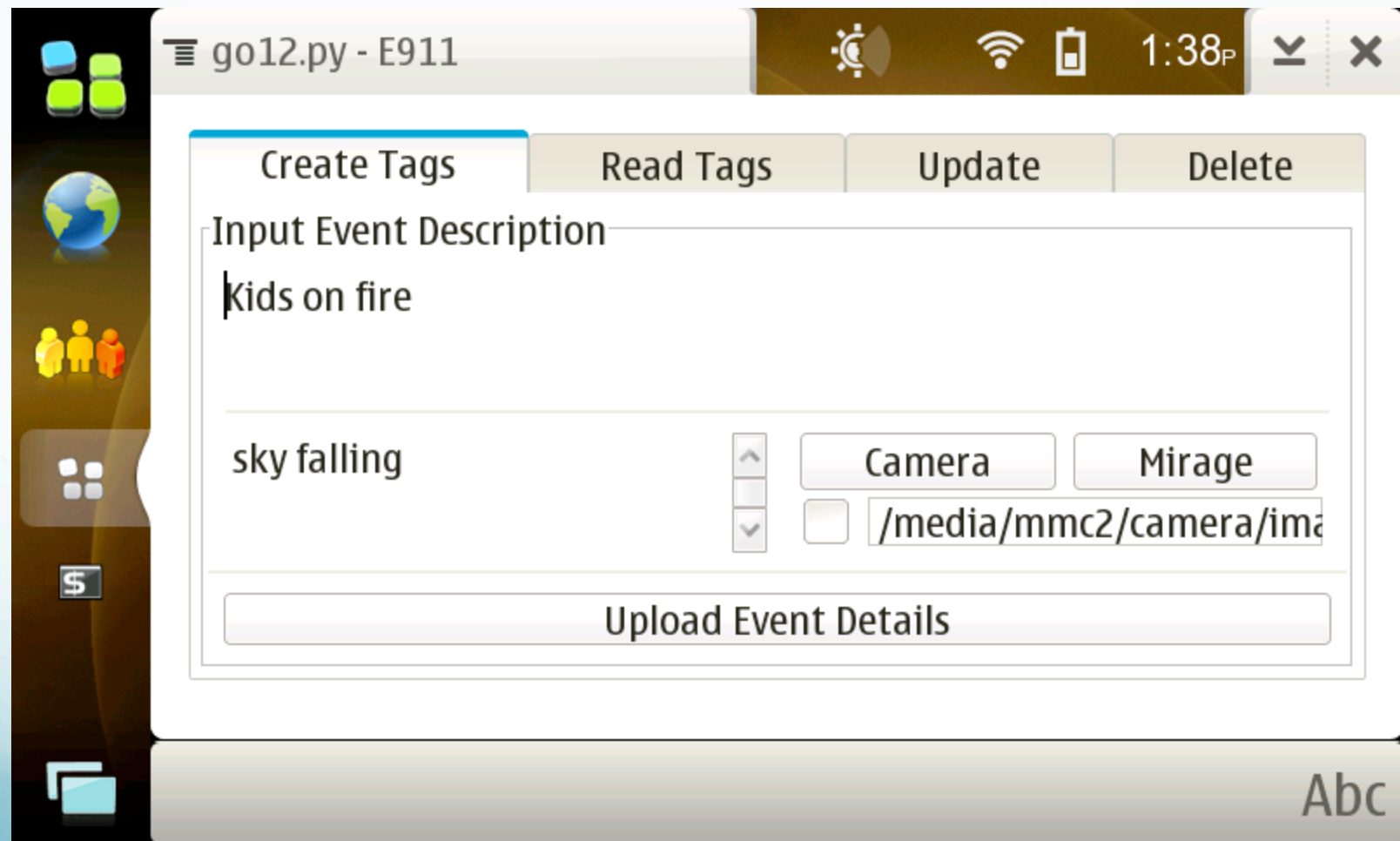
Mobile Dev Challenges

- Mobile Development space is fragmented: Symbian, Java ME, Android, iPhone, Microsoft
 - Applications are typically not platform independent
 - Development tools are not as mature as those for desktop applications
 - Screens are small and device resources are limited
- We chose a Nokia N810 Internet Tablet
It's essentially a resource constrained Linux box

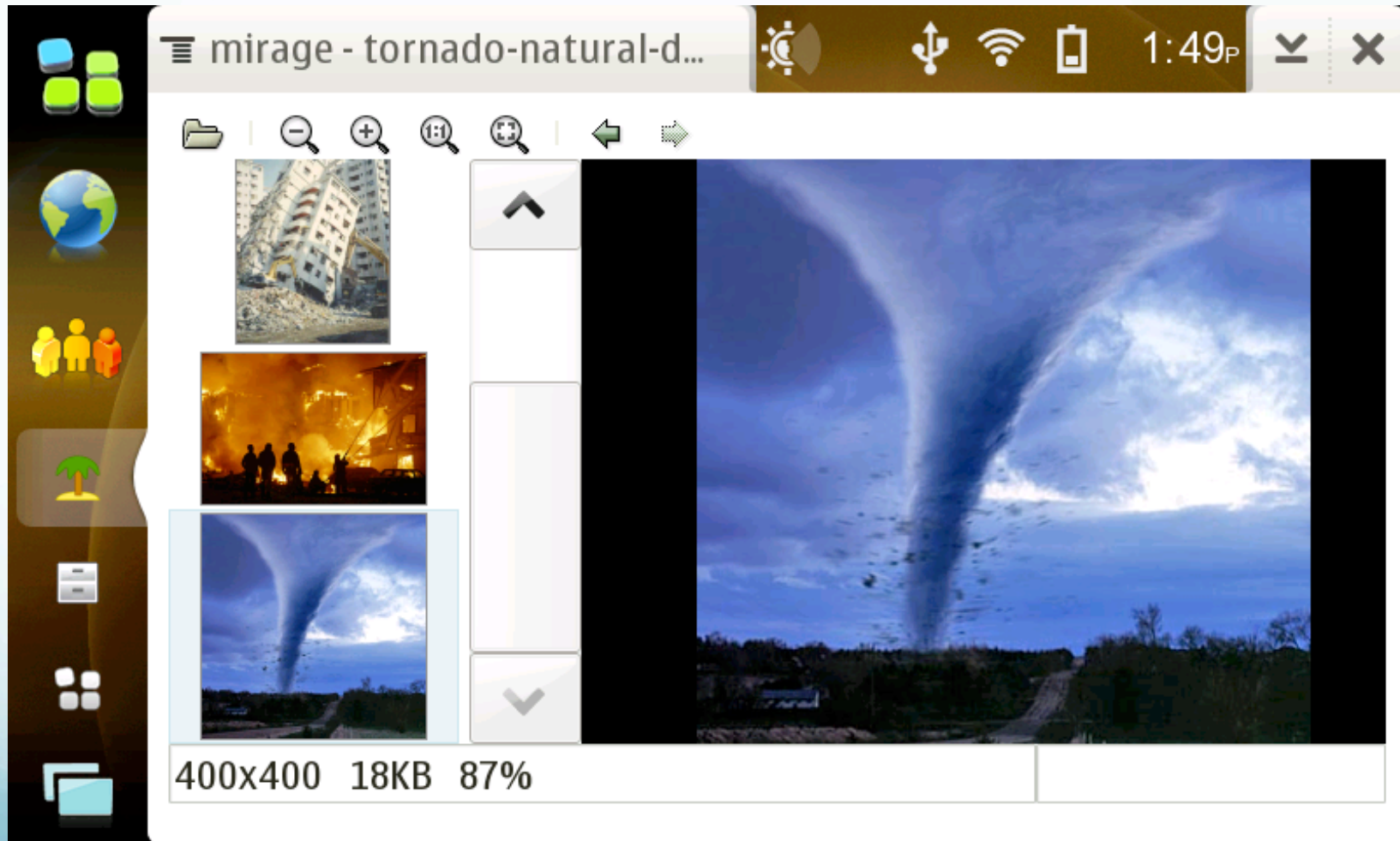
Place In Time Mobile App

- The idea is to enable mobile users to report on events as they're happening
- We needed a front-end that could easily support a camera, image search, file uploads, event description input and tags input
- We decided early on to make use of common Internet resources (e.g. Flickr)

PlaceInTime Mobile App UI



PlaceInTime Mobile App UI

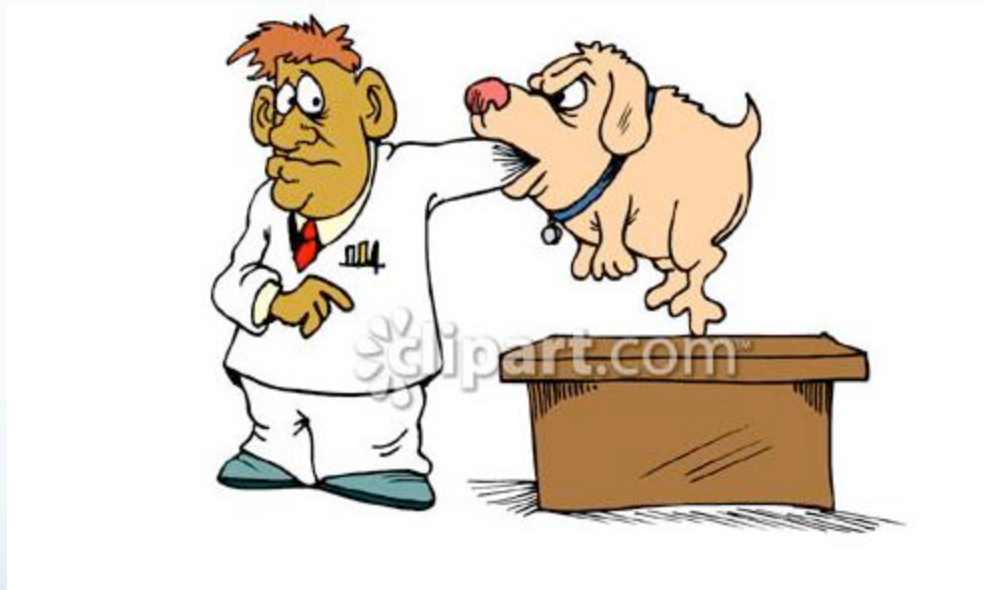


PiT Mobile Technologies

- C – Ugh.
- Python
- Gtk
- Flickr API
- JSON, XML
- Device GPS integration (liblocation)
- Device Camera integration

Issues

- As usual, pay close attention to generated artifacts (they can bite)



Conclusions

- Better integration with social media sites like Flickr, YouTube and Twitter; such as query events from the front-end more directly
- Web service could support other media types (video, audio, etc) and post them to relevant sites
- Better search functionality on the front-end
- Be able to show multiple episodes at same time



Thank You

Questions?