

DynMap App Contest Presentation

This is a rough outline that was the basis of my DynMap presentation, given on March 21, 2013 as part of the inaugural Dyn app contest. The [main DynMap page](#) has more complete details on features, architecture, and installation.

Intro

- Most of you know I work on the DynECT Email product
- Always had an interest in customer interaction, working on websites before Dyn, etc.
- Email can be boring – “we take it in, we send it out”
 - But there's so much more – account setup, SPF/DKIM, bounces, complaints, deliverability, etc.
- Beyond customer customer interaction, recipients of emails also interact:
 - **opens** – when they open it, and for how long
 - **clicks** – when they click a link in an email



Demo

- <http://dynmap.mht.dyndns.com>
- DynMap displays a live, real-time view of recipients throughout the world that are opening and clicking email sent through DynECT Email
- Feature walkthrough
 - checkboxes to enable/disable view of open and click events
 - drag map to pan from area to area
 - zoom buttons; also double-click to zoom
 - map viewers shows how many people are viewing the map
- Try it out!
 - requires VPN connection
 - limit of 25K map points per day at this point (free plan), so it may not work

Architecture

- Data collection
 - open/click tracking server (nginx) logs
 - [Flume \(ng\)](#) exec source to tail logs into Flume channel
- Data flow
 - *Custom Flume sink* to transfer data to a [Node.js](#) daemon for conversion to lat/lng
 - Same Node.js daemon sends lat/lng data to map clients
- Map client
 - [Google Maps JavaScript API](#)
 - [socket.io](#) library to achieve publish/subscribe connection, allowing server to publish (push) map events to clients that are subscribed, instead of map clients having to poll
- Currently using Python script to replay logs due to apparent Flume batch size bug
- System is architected to allow for some scalability
 - Different Node.js daemons for each type of map event

- Flume is natively scalable, taking events as they come from anywhere and routing them to many places
- Event-based model ensures data is sent only when there is data to send (no polling)
- We could even display per-customer activity on a map, as a report! (current or historical)