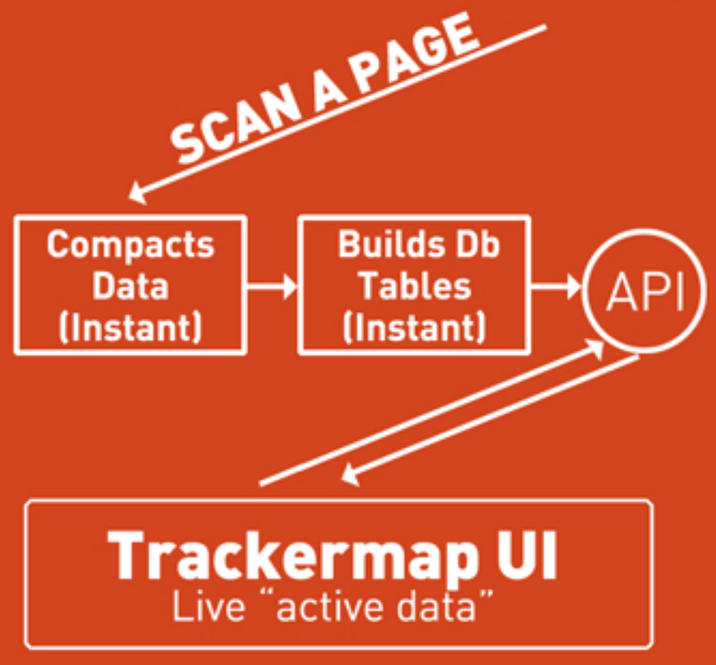
TRACKERMAP - TECHNICAL DEFINITIONS

DATA

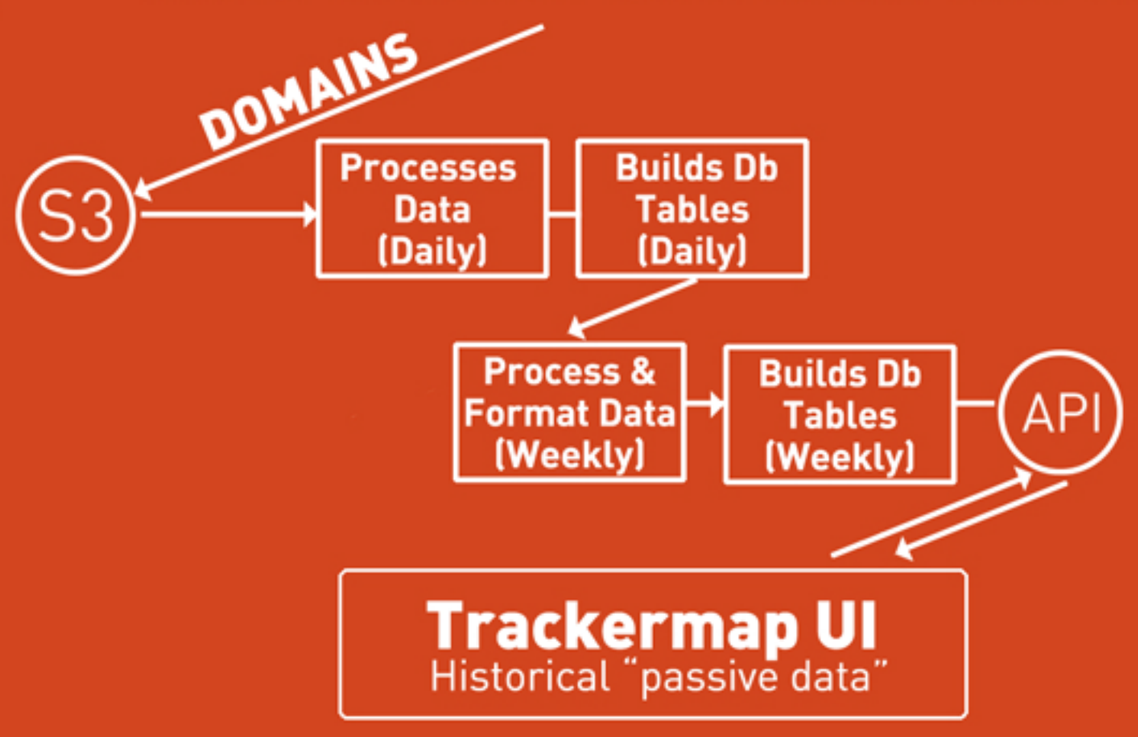
LIVE based on GADGET (.NET)



1. Gadget scans the webpage for all element activity.
2. An instantaneous process compacts the results (only one data point, no aggregation).
3. Another process then pushes the data into the BS Db (BS stands for Balls & Sticks), creating a specific structure of connectors and nodes.
4. From the Trackermap UI*, an API requests this data (based on user domain selection) and retrieves the "Balls and Sticks" data.
5. The resulting data is formatted and rendered into the Trackermap UIwith nodes and timelines.

* The UI uses D3 Library to render and display the map in circular nodes and timelines.

HISTORICAL based on SLEUTH* (Haskell)



1. Evilabs* scans data from target sites and stores the results in files on Amazon S3*.
2. A process reads these files and creates a daily set of aggregate records in MSSQL tables.
3. Weekly, a process reads the daily tables and creates the final aggregate Trackermap records in MSSQL Db. This is often referred to internally as "Balls and Sticks".
4. From the Trackermap UI, an API requests this data (based on user domain selection) and retrieves the "Balls and Sticks" data.
5. The resulting data is formatted and rendered into the Trackermap UI with nodes and timelines.

* S3 is a file storage system from Amazon, and it works across all regions.
* Evilabs is the older sleuth based scanner system written in Haskell. We currenly scan about 630 million pages with the Evilab systems on a 16 node cluster with 300+ scanner clients.
* Gadget is the new scanner engine which is more accurate and slated to replace Evilabs early Q2.

OPTIONS

LIVE

Scan a Live Page

The scanner instantly crawls *one* webpage, the system aggregates and formats the data, the API pulls in that data and renders it into nodes and timelines.

Scan a Tag

When Tag code is typed (or copied and pasted), the system loads and executes the tag on a hosted webpage.

Scan From

When a location is chosen, the system scans from that location and the systems serves its page content as if the user were physically in that geographical location.

HISTORICAL

Domain Group

Currently derives from Evilab data.
The API retrieves matching Data from the MSSQL "Balls and Sticks" Data and displays it together in aggregate form.
Because multiple domains may be aggregated together, the data may lose some accuracy.

Domain

Works the same as Domain Group - derives from Evilab data.
Rendering a single domain typically gives more directly relatable information in the Trackermap format.

INTERFACE

LIVE & HISTORICAL

Prevalence view

When scanning elements (tags, cookies, etc.), instantly or within a timeframe, the system requests a *count*, looking for frequency.

Tag Vendor Type (Tag Purpose)

Created within the BUGS Db, a pattern matcher reviews each company application, which has a "type ID ", and identifies and matches these IDs to one of 7 internal categories.

Isolations and Tag Chain

Double-click a node to isolate and initiate a 'vendor filter', which highlights only the nodes related to the original call and its direct children.
Tag Chain looks at child tag load numbers and presents the most prevalent chain, along with average latency numbers for each tag along the chain.
Note: Live scan has a more dependable tag chain, as there is less data to filter with a single webpage.

Latency view

The system measures the time between each tag being called to when it fully ***loads***, then calculates the average.
Note: There are no thresholds for latency calculations, meaning the system does NOT throw out any tag that takes too little or too long to load.
Note: The size of the nodes are based on the latency number.

Relationships and Resource Weight

The Relationship (direction of communication/resource request) is defined by grey arrows and the size of the arrow indicates the amount of point to point communication happening.
Resource Weight indicates the size of the node, and is the count of how many times the tag is called within the (instant or weekly) scan.

Filters

Non-secure Tags: When an HTTPS page has an HTTP tag, it can throw a browser warning as a potentially non-secure page.
The system runs a "protocol change check" to identify these tags.

New Tags (Historical only)**:** When scanning, the system marks any new tag that wasn't in the system the previous week.

VIEWS

Timeline (Live only)

Run through the same API process, but rendered in a different visual way.
This view shows *every* instance of a tag, regardless of redundancy of vendors.
Note: Timeline works in the same way as Google Dev Tools when you press F12 in Chrome.
Named *Call Tree* in the new version, and shows communications back and forth between vendors before a tag is loaded.

Tree (Historical and Live)

Same UI to/from API process, but presented in a flat, tree-view format. Content is exactly the same as in TrackerMap.
Note: Named *Node Tree* in the new version.

Note: Trackermap looks at *load* time of a tag. Once a tag is loaded, then called to execute,it could call additional tags, which load across or after a page has loaded. As opposed to showing this as one tag, the new Call Tree shows this entire chain of staggered events, providing higher counts per tag and more accurate latency numbers.

EVIDON from crownpeak