



From Bandwidth to Beacon Detection, Prism and Touchpoints

George Jones
Paul Krystosek
Sid Faber
SEI CERT NetSA



NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

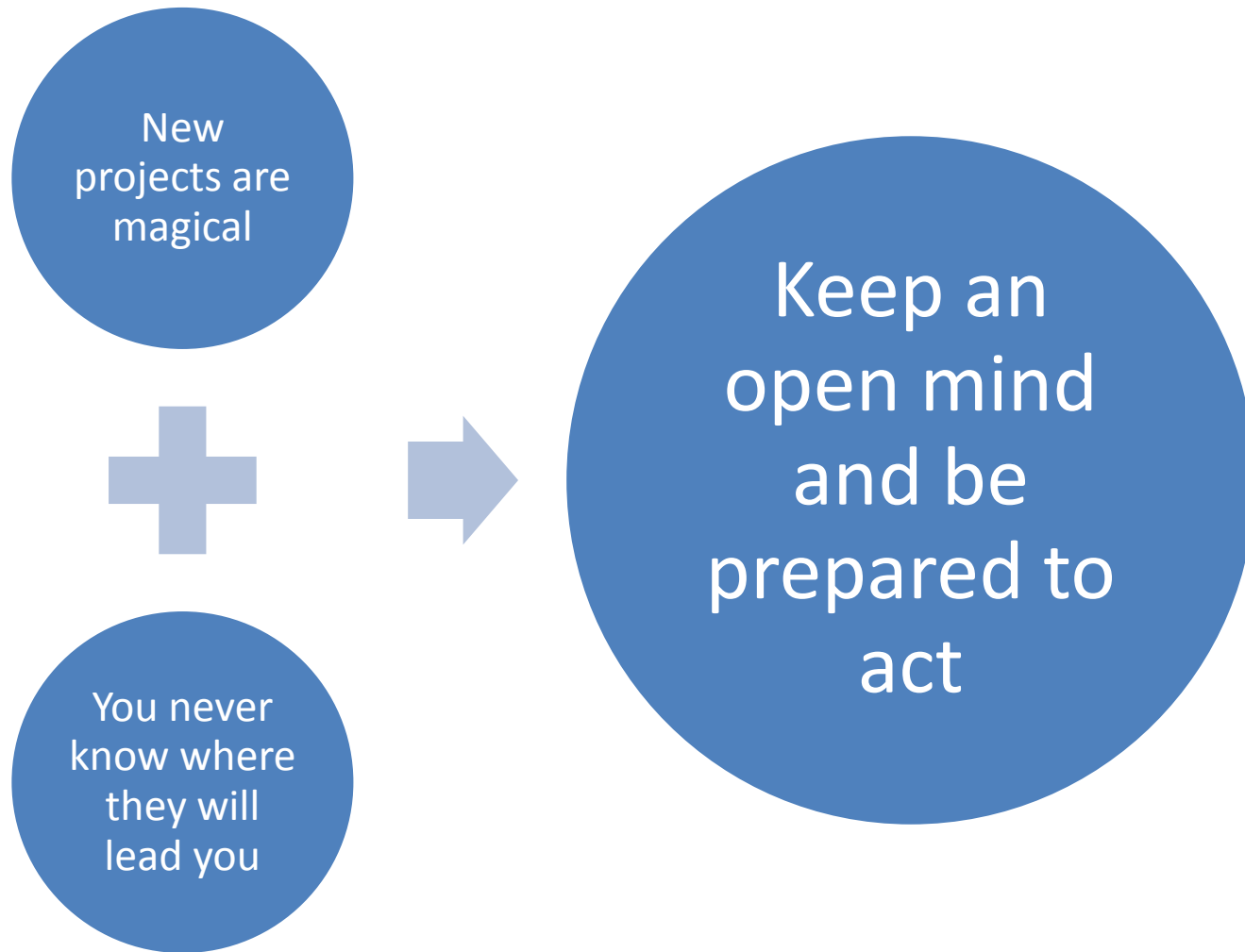
Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

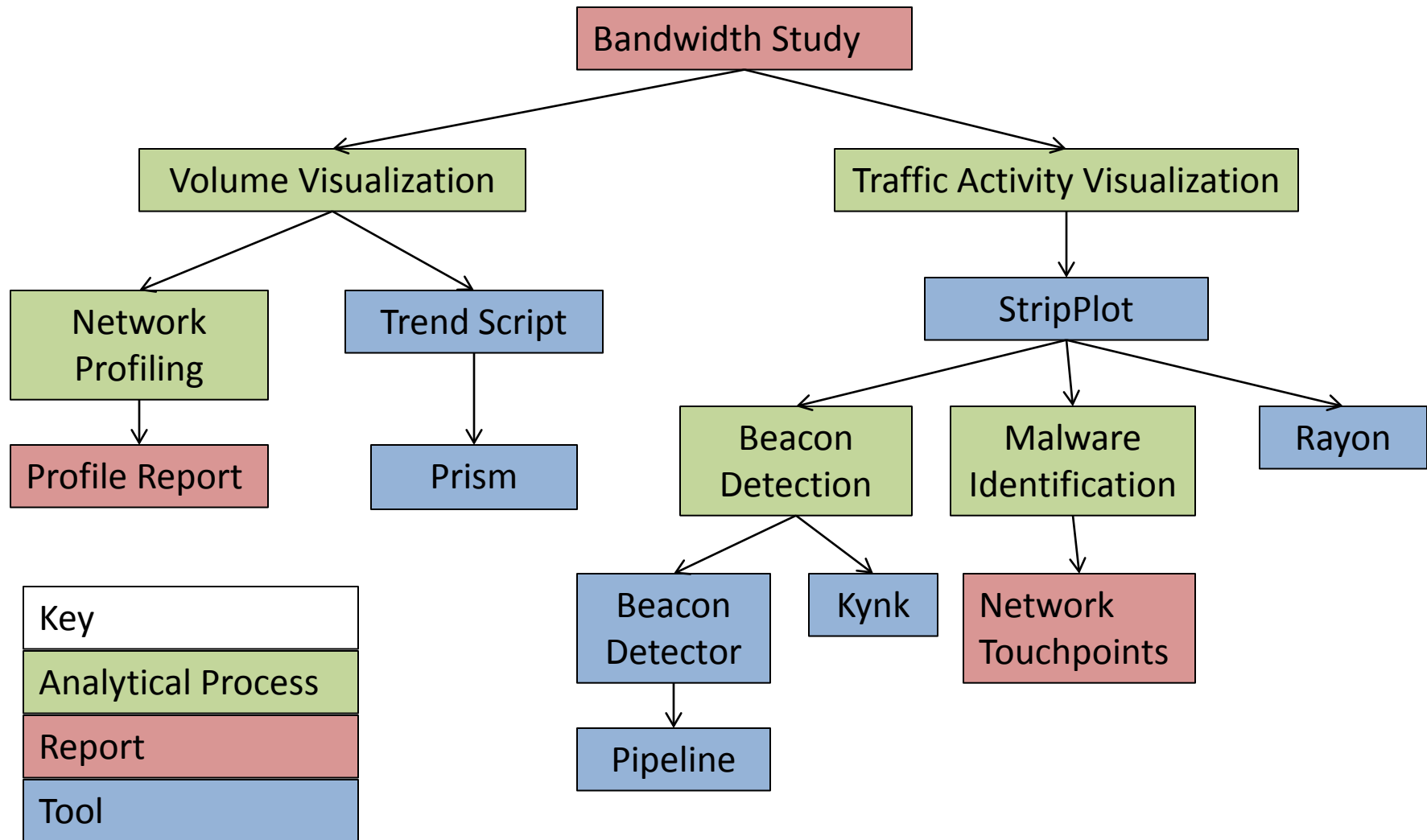
This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.



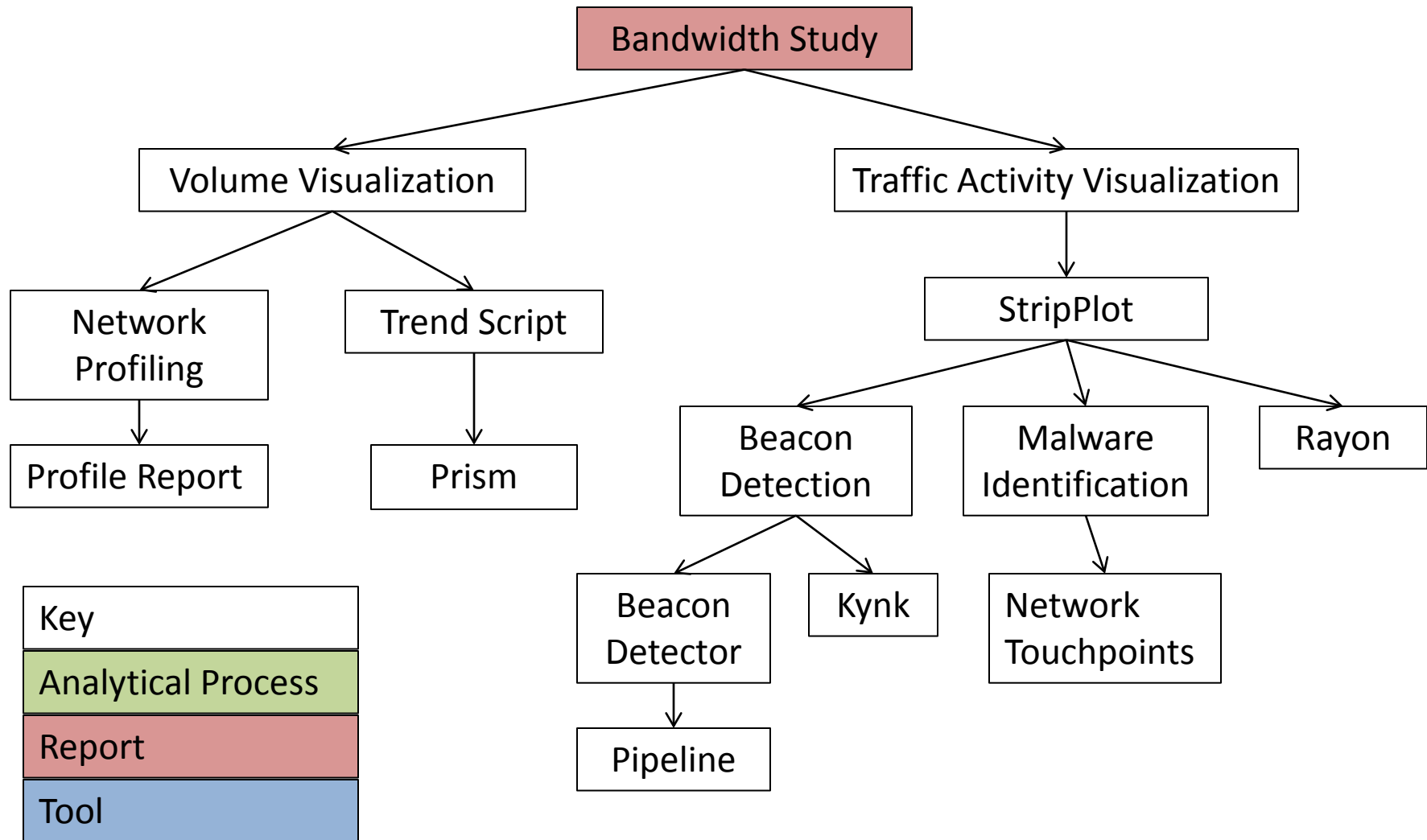
Introduction



Overview



The starting point



The Bandwidth Study

Once upon a time...

There was a network that everyone thought was dirty.

They planned to get some sensors in place...
but all they had for now was flow.

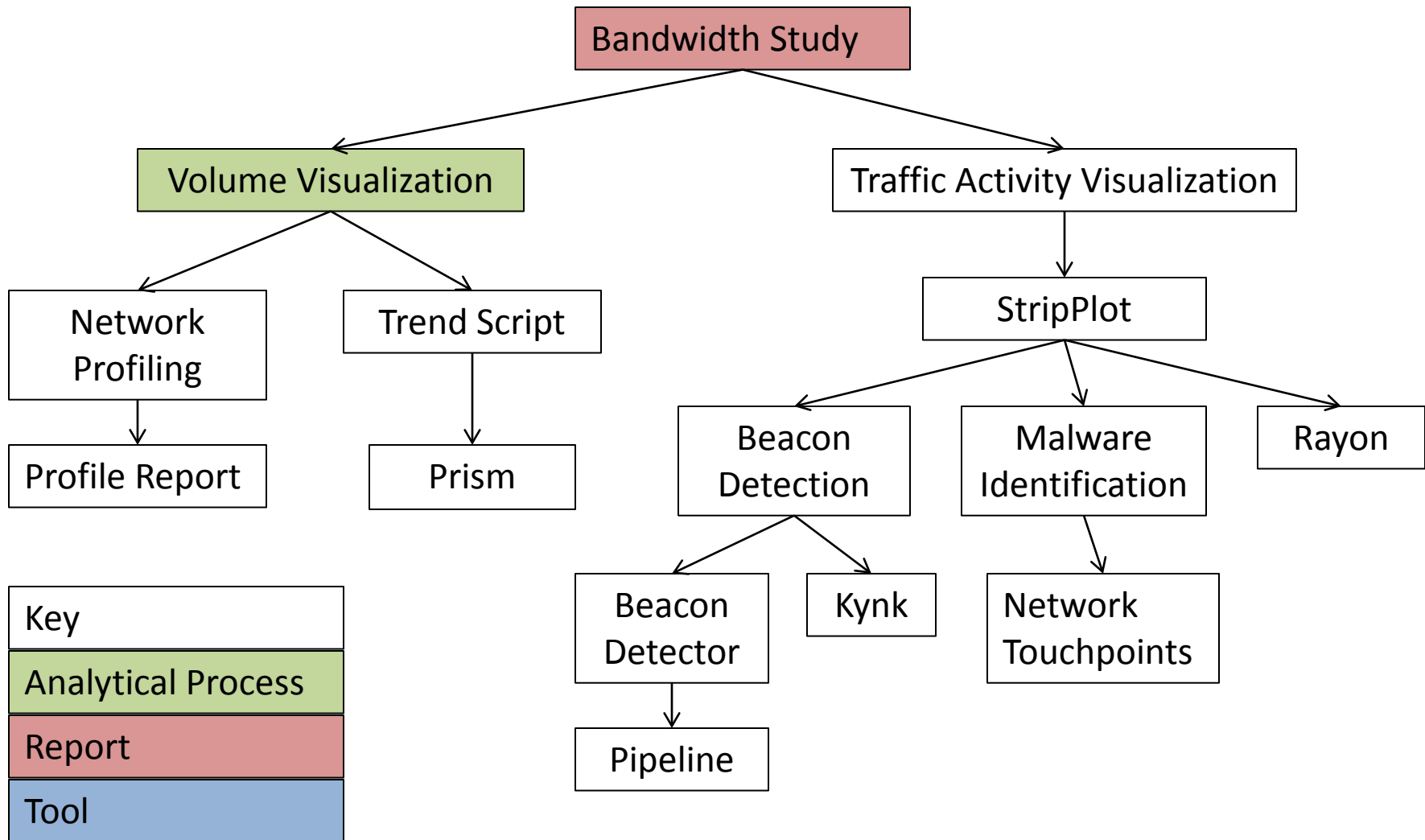
What could be done to keep them safe until sensors were deployed?

This is where our story starts

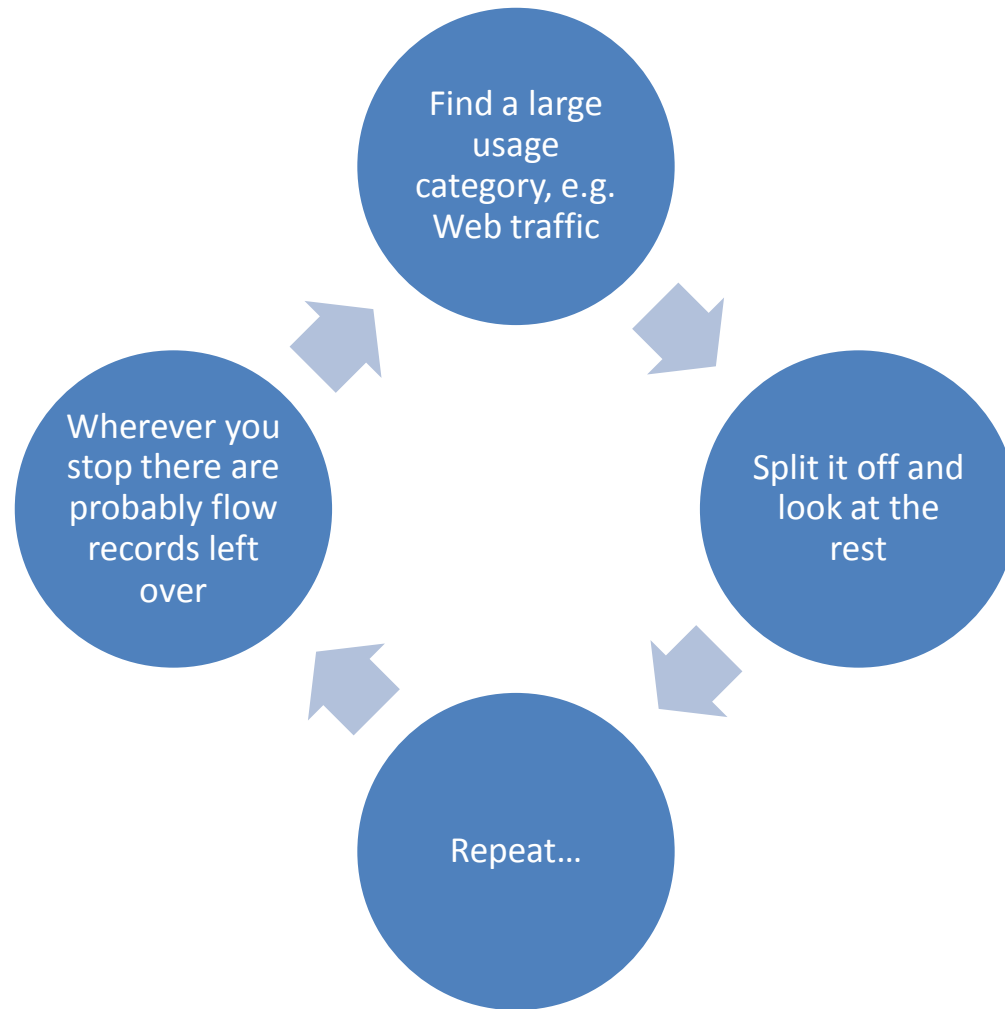
From there it meanders hither and yon



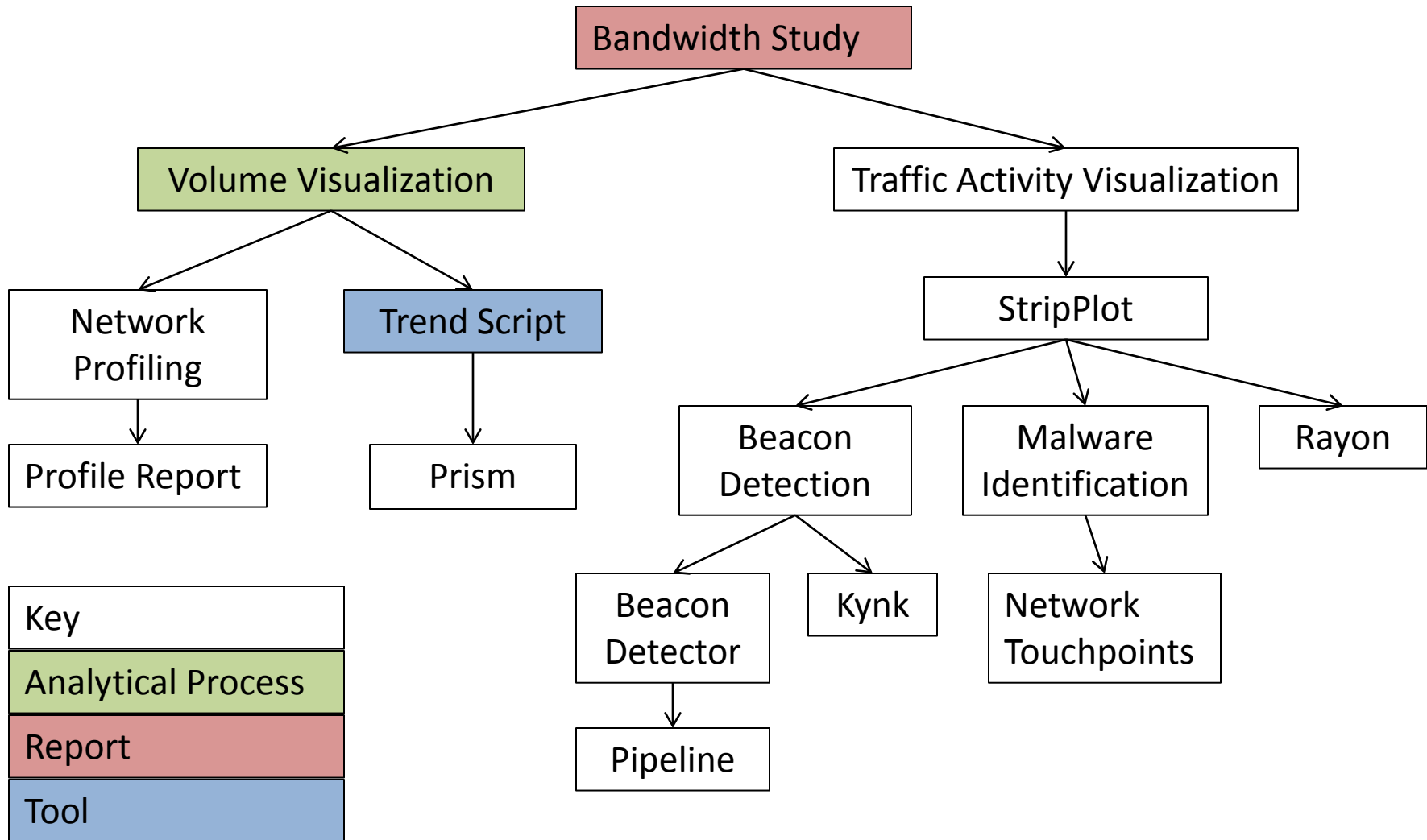
Overview



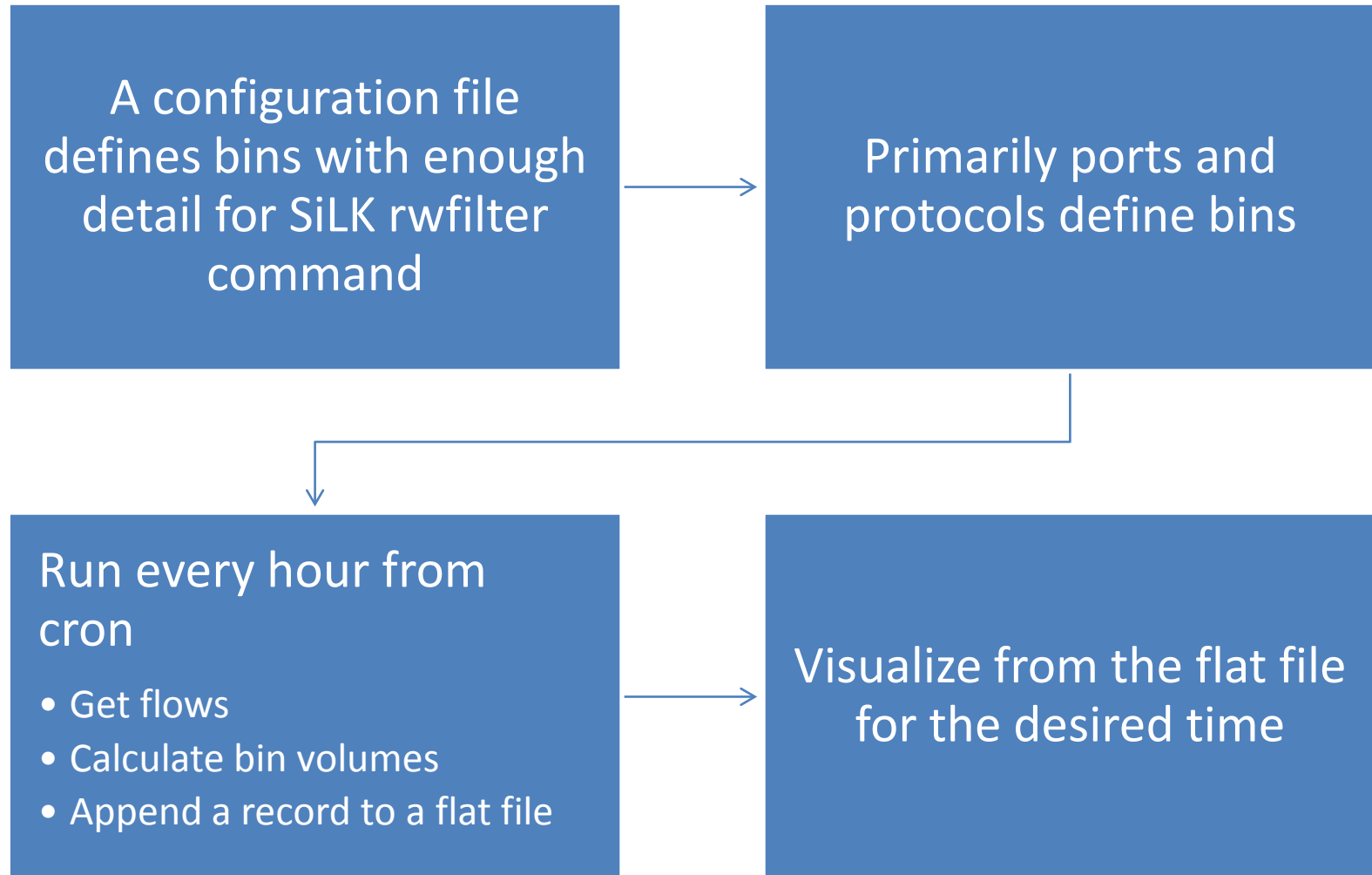
An iterative Process



Overview



The Trend Script is born



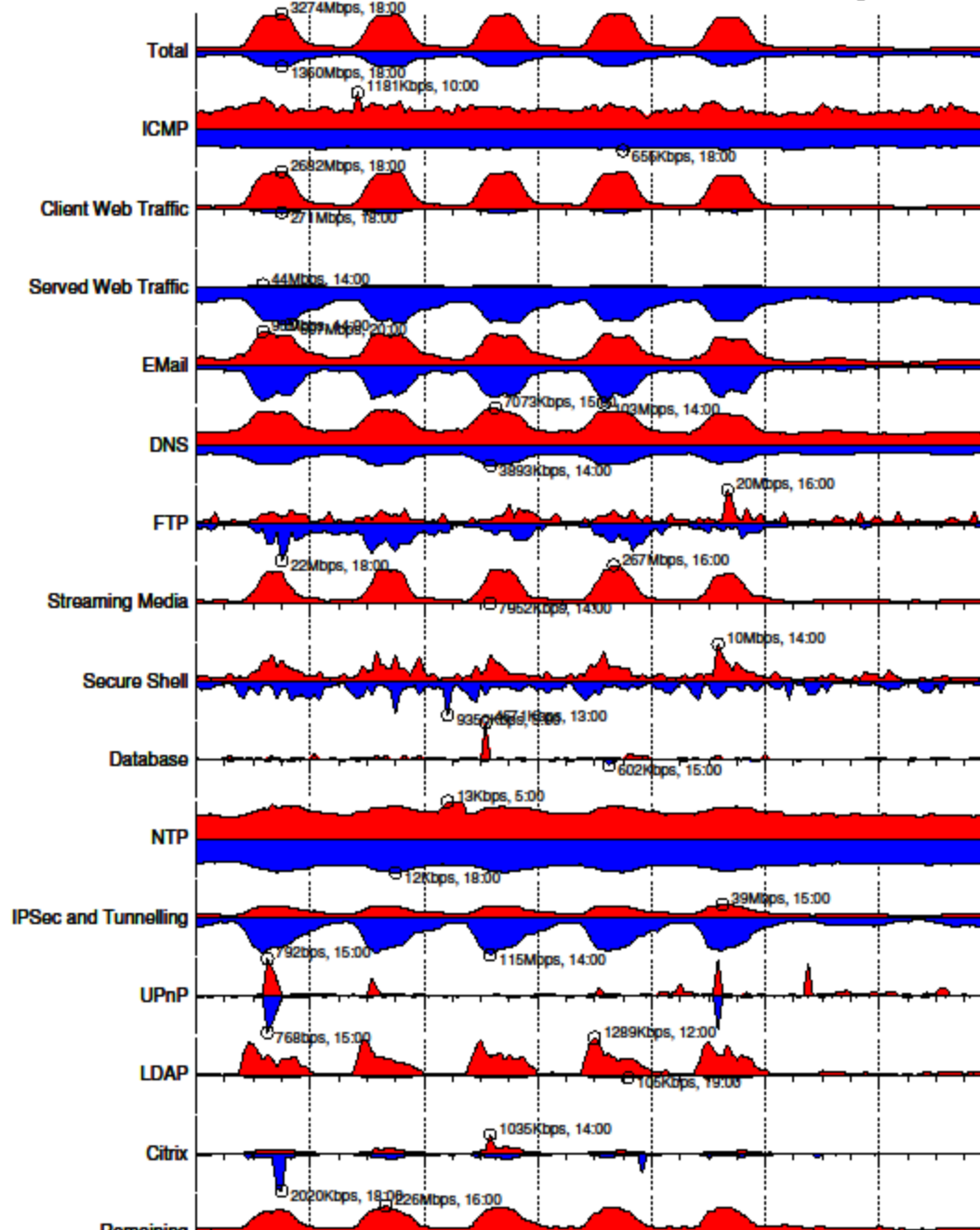
Sample Trend Script Configuration

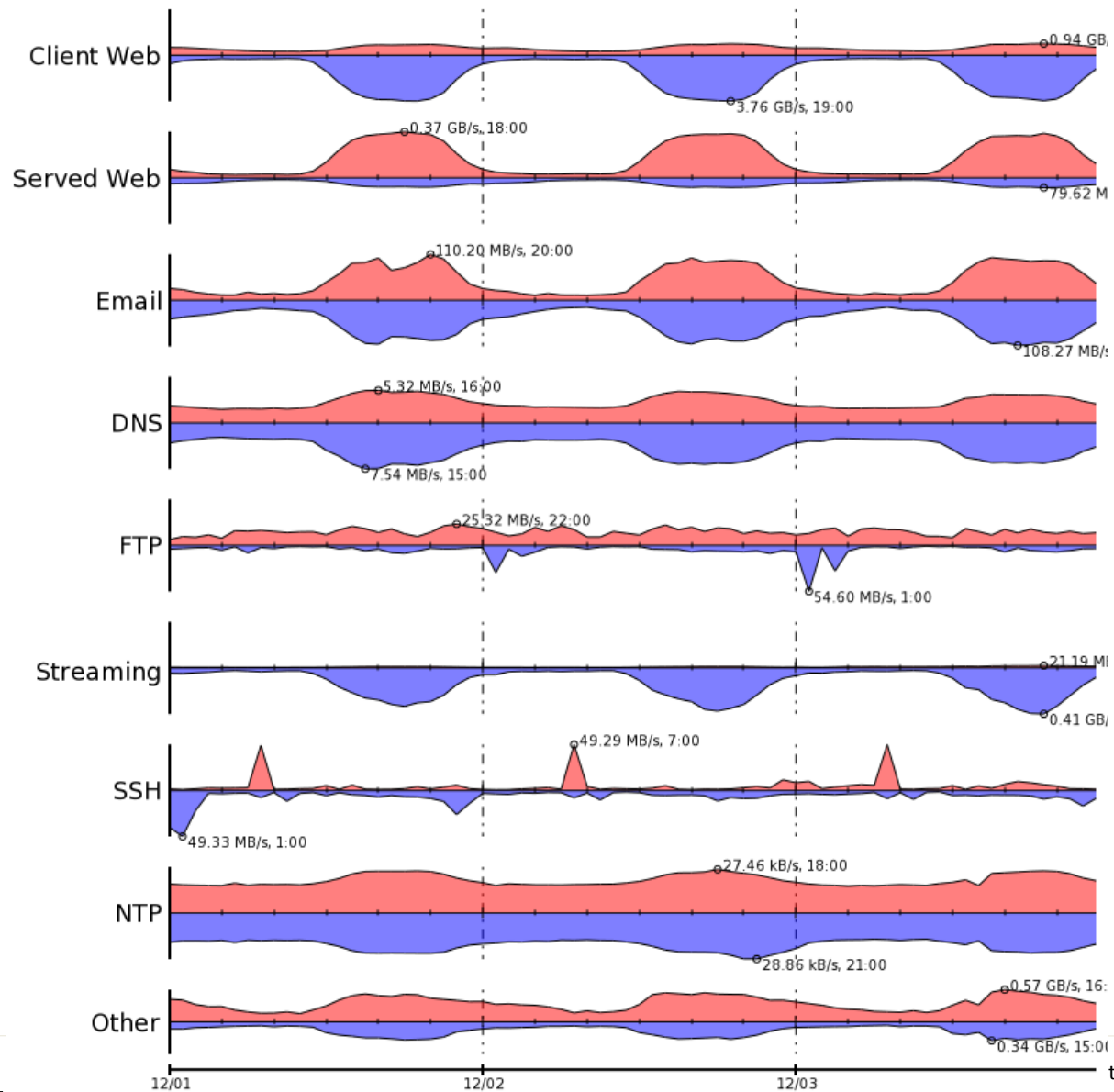
```
[bin]
name: http-client
title: Client Web
filter: --protocol=6
out-filter: --dport=80,443,8080
in-filter: --sport=80,443,8080
```



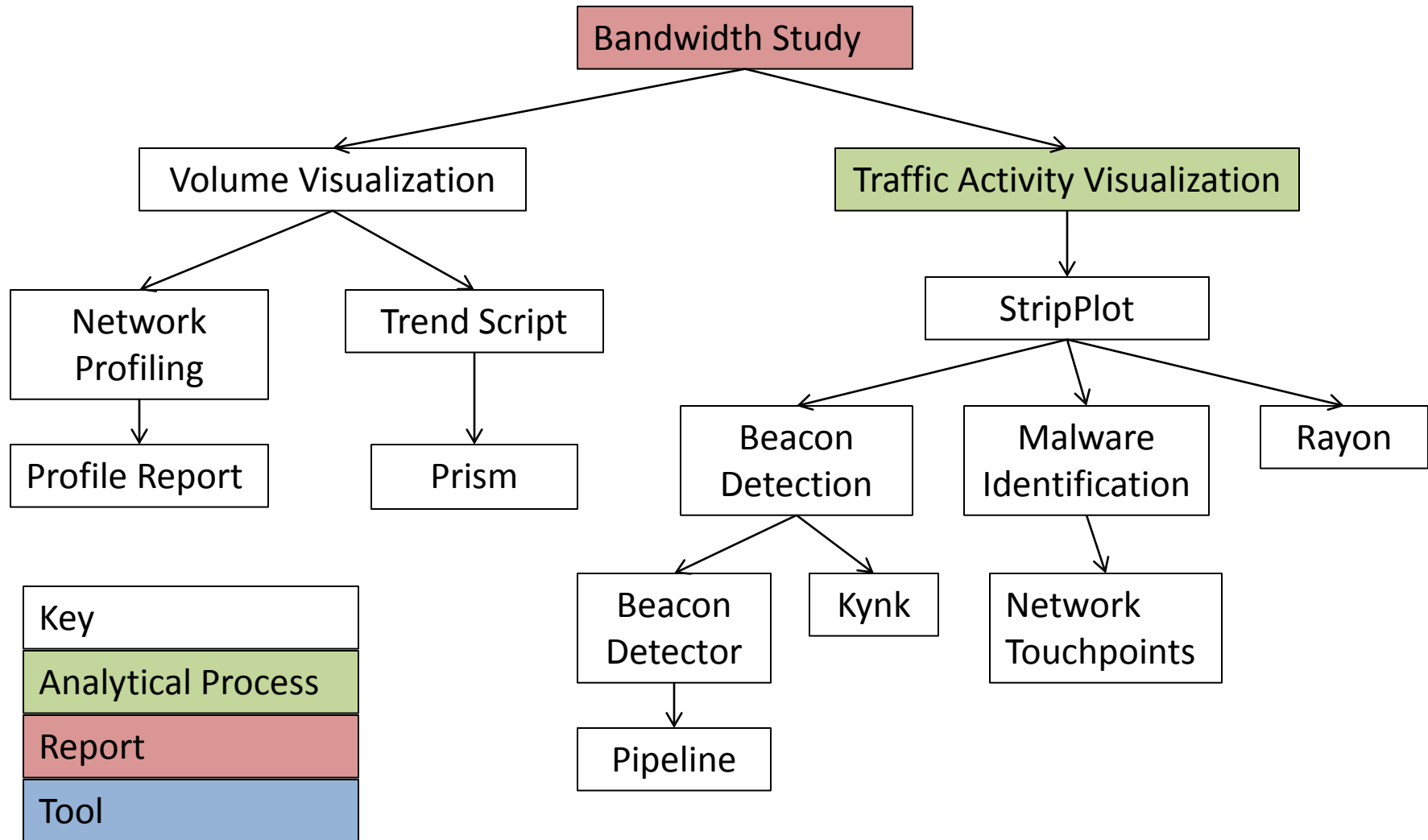
One Week of Activity on All Sensors

● Outbound bytes
● Inbound bytes





Overview



Traffic Activity

The activity in the bins is fairly well known

The “left over” flows, less so

What is happening “at the edge”?

Looking at flows by hand is tedious

It’s hard to program looking for the unknown

That means, it’s time for...



Flow Activity Visualization

We want to find “interesting activity”

But interesting means different things to different people

- “May you live in interesting times.”

Chinese Curse

- “Only accurate rifles are interesting.”

Colonel Townsend Whelen

- “The only interesting answers are those that destroy the questions.”

Susan Sontag



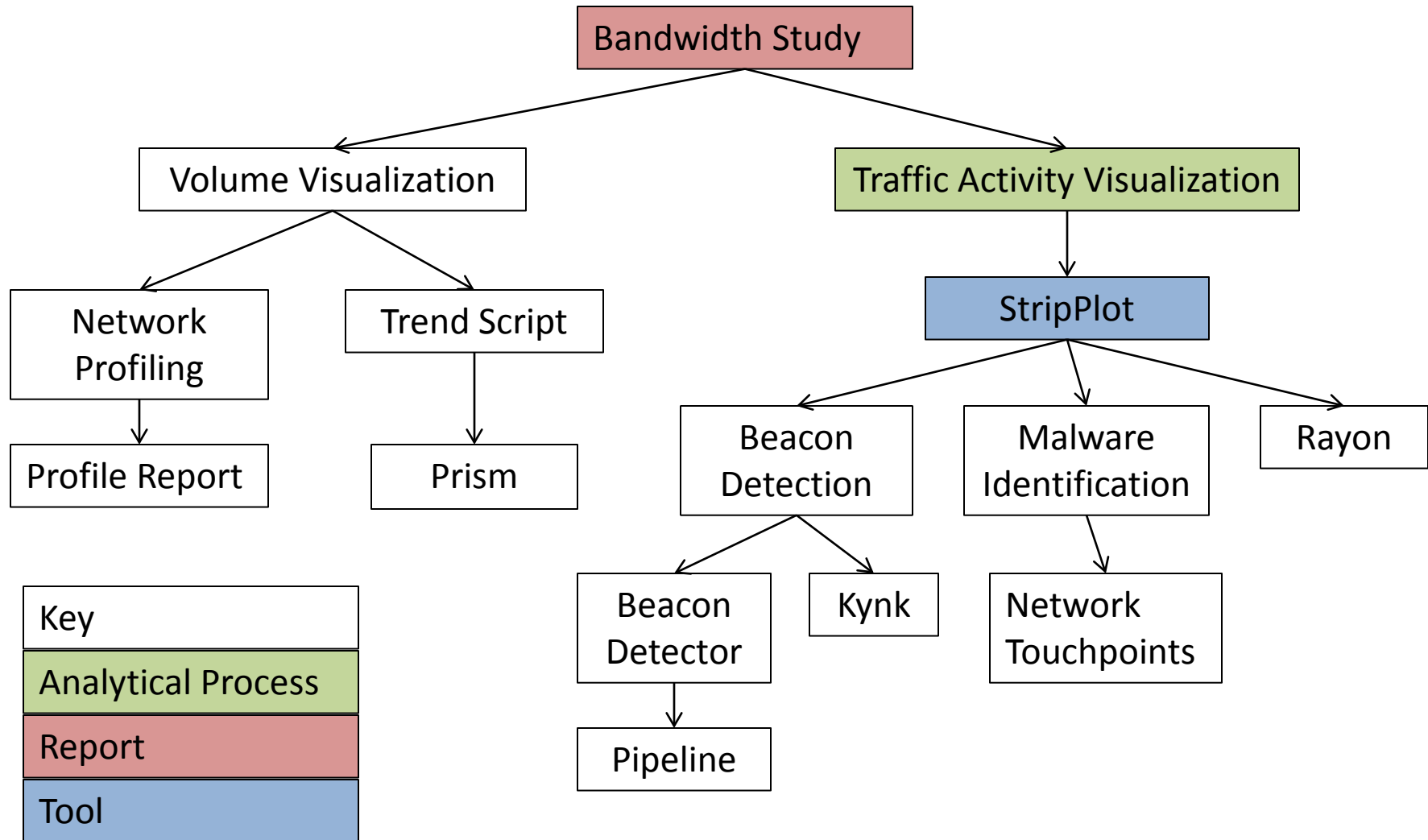
Flow Activity Visualization

Goal: produce a self-maintaining network profile

- Categorize and display activity
 - Stuff we know about: Email, Web, DNS...
 - And everything else
- Need a mechanism to permit the analyst to examine “everything else” aka leftovers
- Too bad about the “self-maintaining” part



Overview



StripPlot “enables the eyeball”

Get a good idea of what a particular IP addressing is doing

See how a port is used

Streaming video and audio are immediately apparent

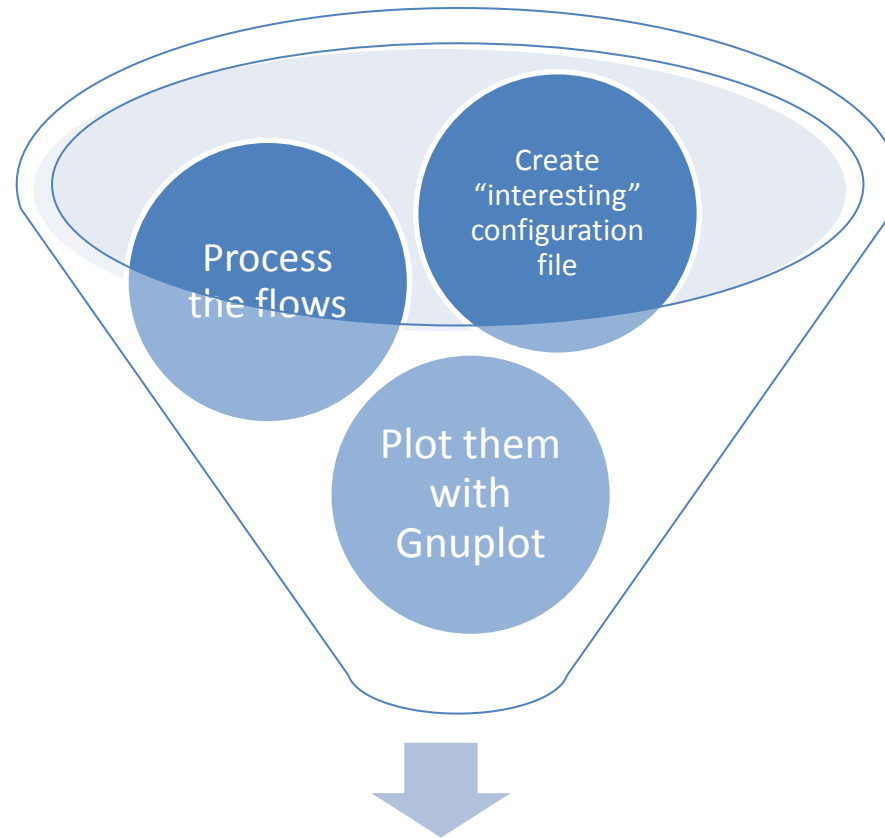
Make Beacons stand out

For more information on StripPlot see:

- http://www.cert.org/flocon/2010/presentations/Faber_StripPlots.pdf



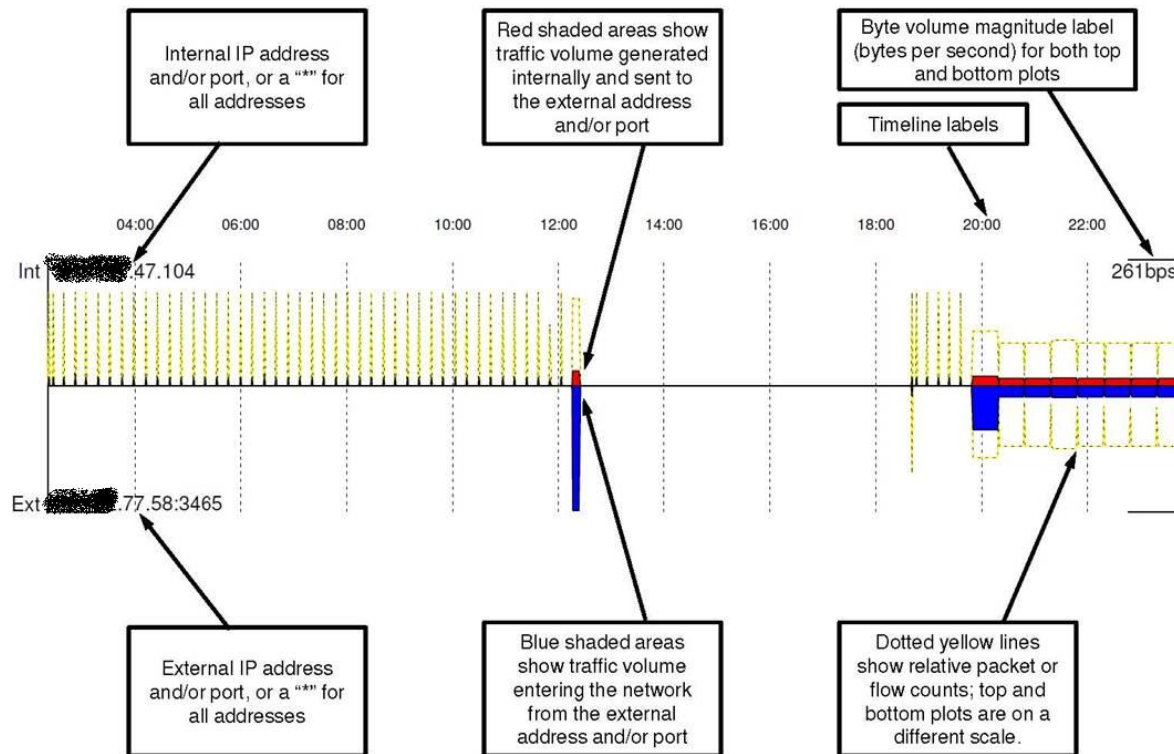
The StripPlot Process



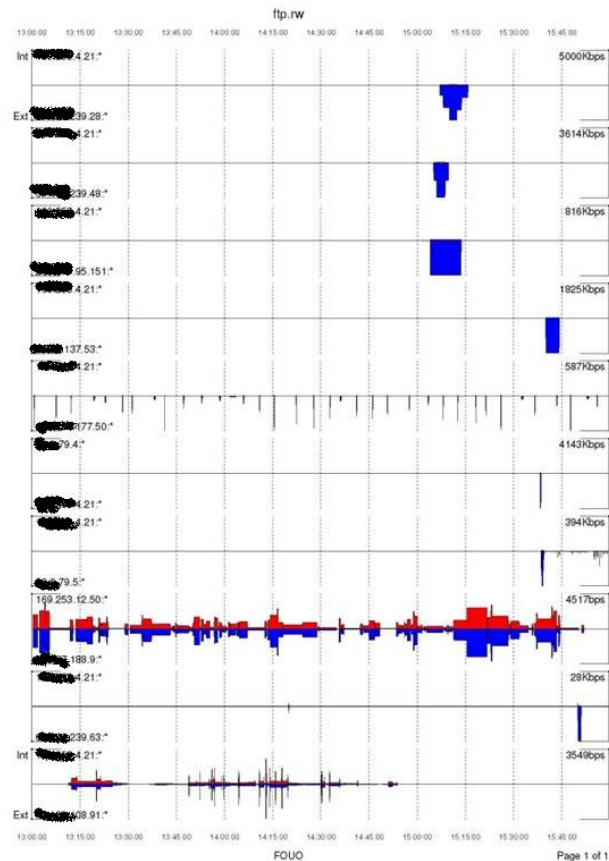
StripPlot Graphic



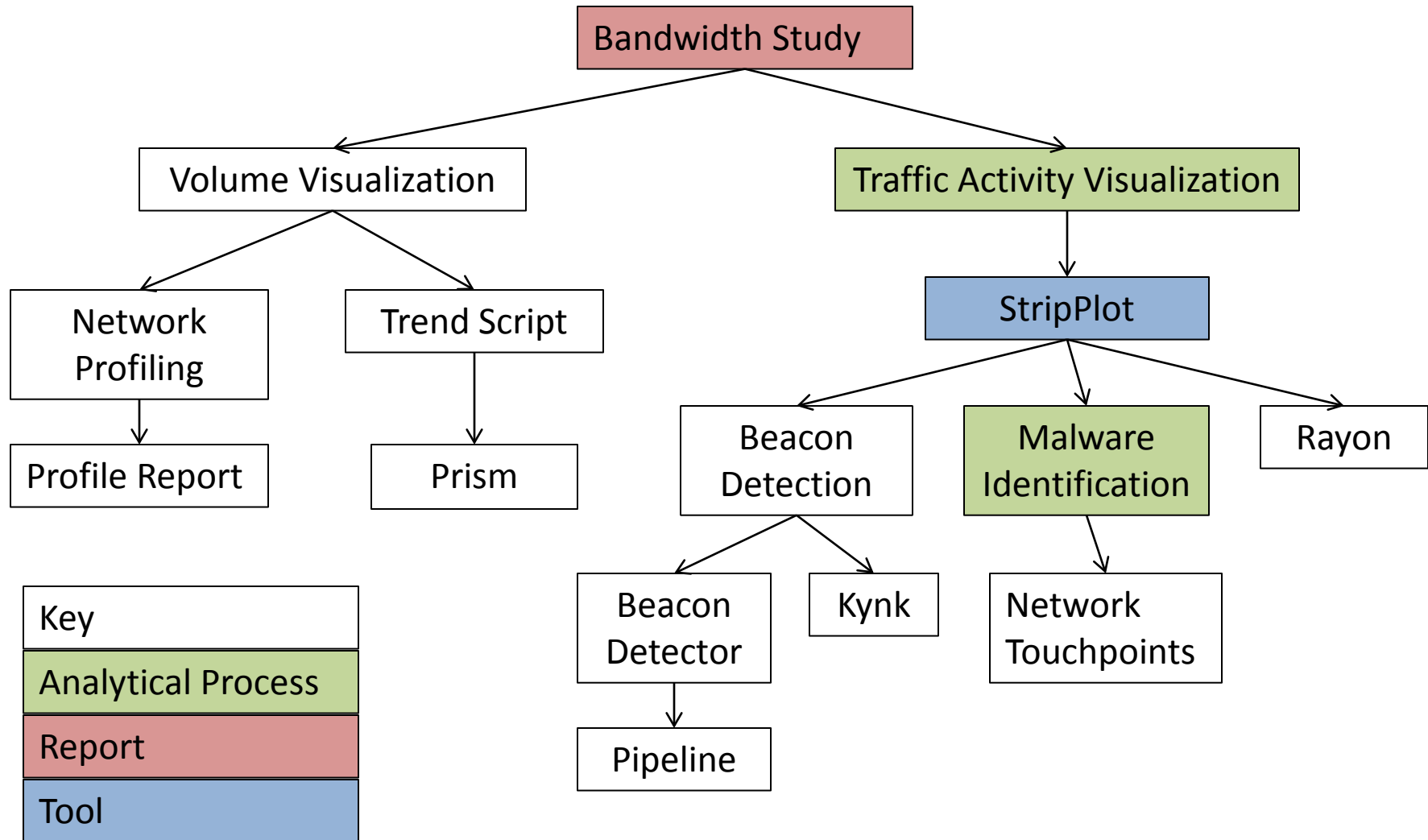
How to Interpret StripPlot



Sample StripPlot



Overview



Finding Malicious Activity

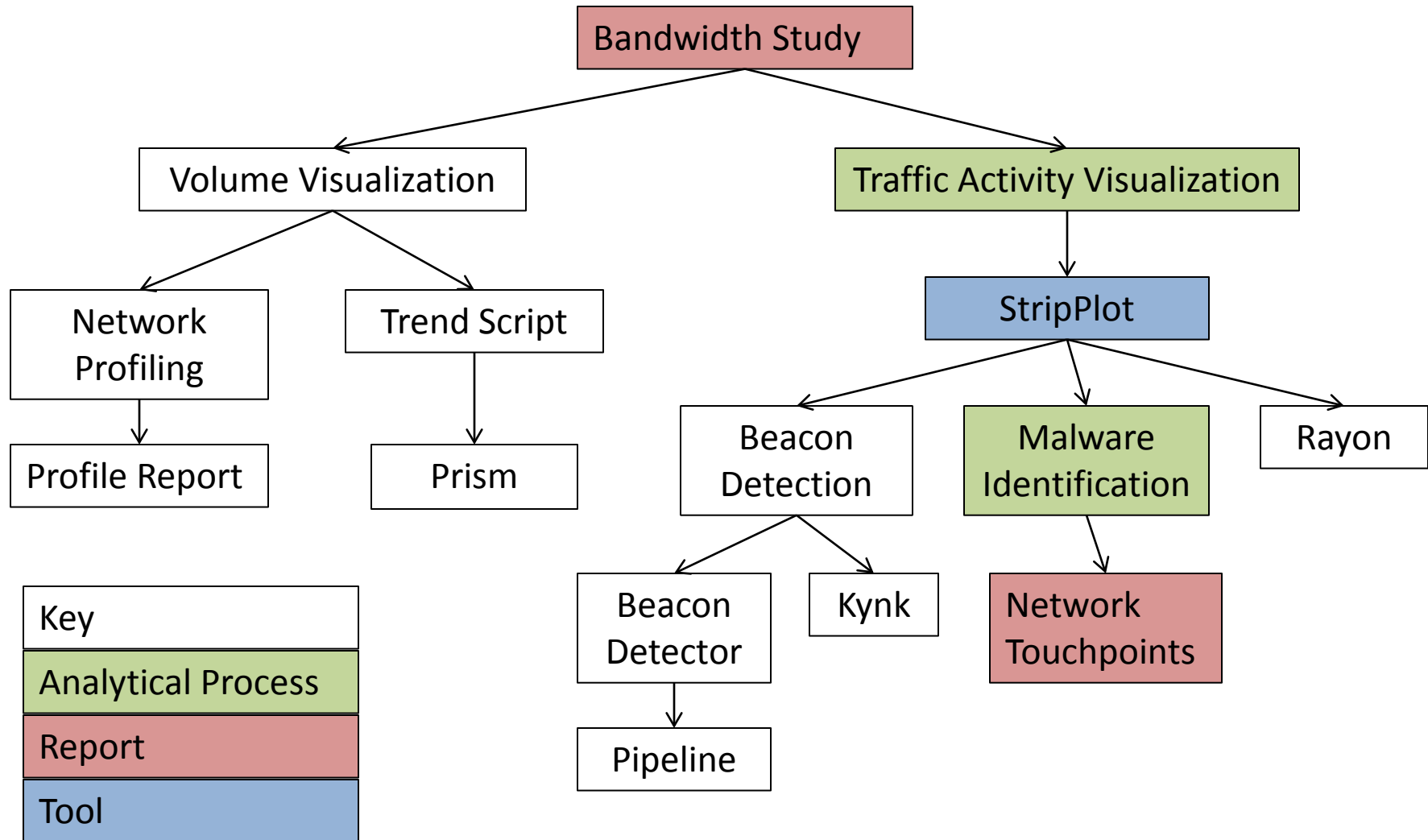
Malware Team to NetSA Analysis Team:

- “You might find this interesting”

The visualization in StripPlot made it easy to spot the interesting behavior



Overview



Spin off the Network Touchpoints Project

Find network indicators in
malware



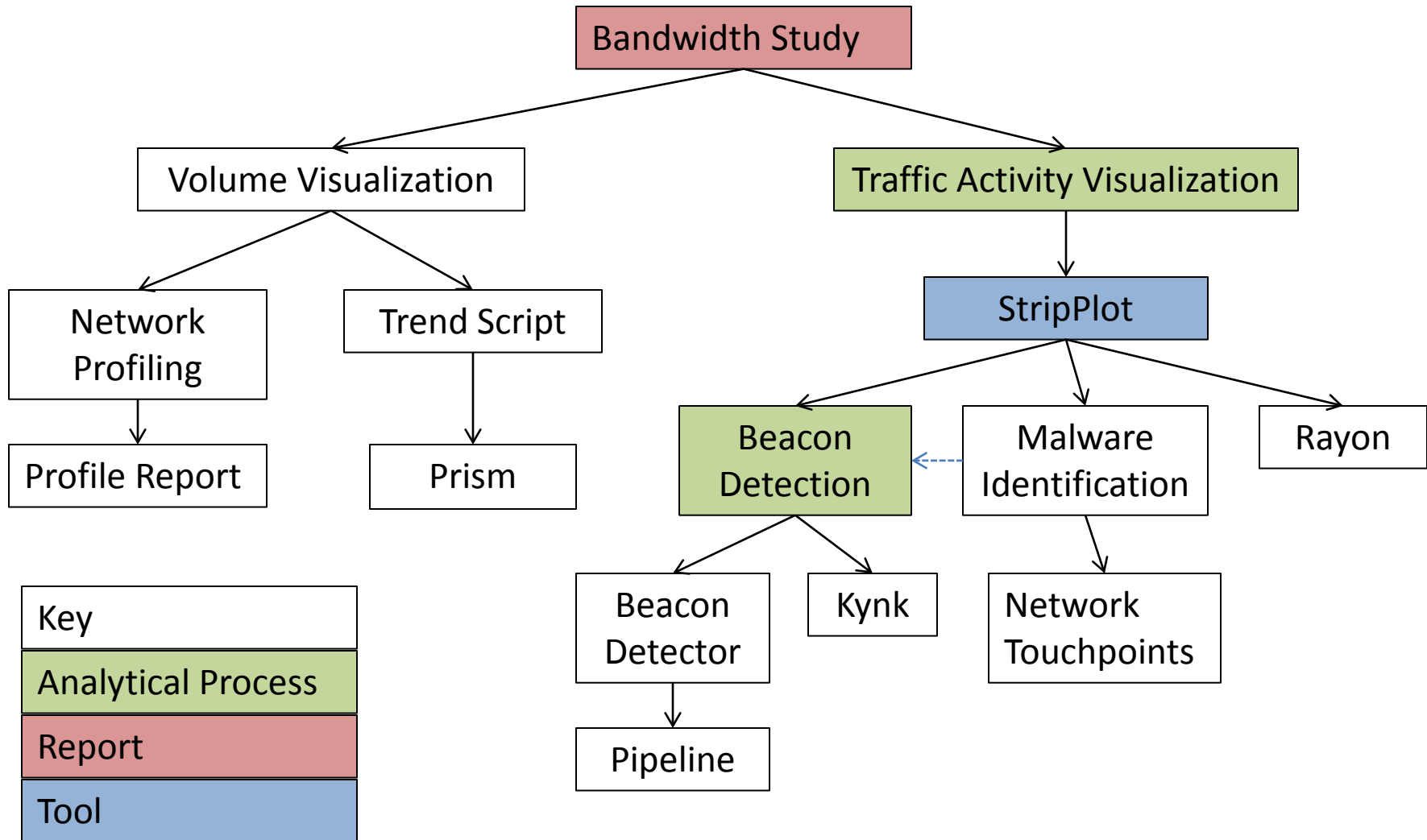
Find the indicators in
Flow



Characterize and Report



Overview



Beacon Detection

StripPlot “enabled the eyeball” to see botnet nodes phoning home

We even saw a handoff from one C2 host to another

Beacon detection attempts to “replace the eyeball”



Beacon Detection

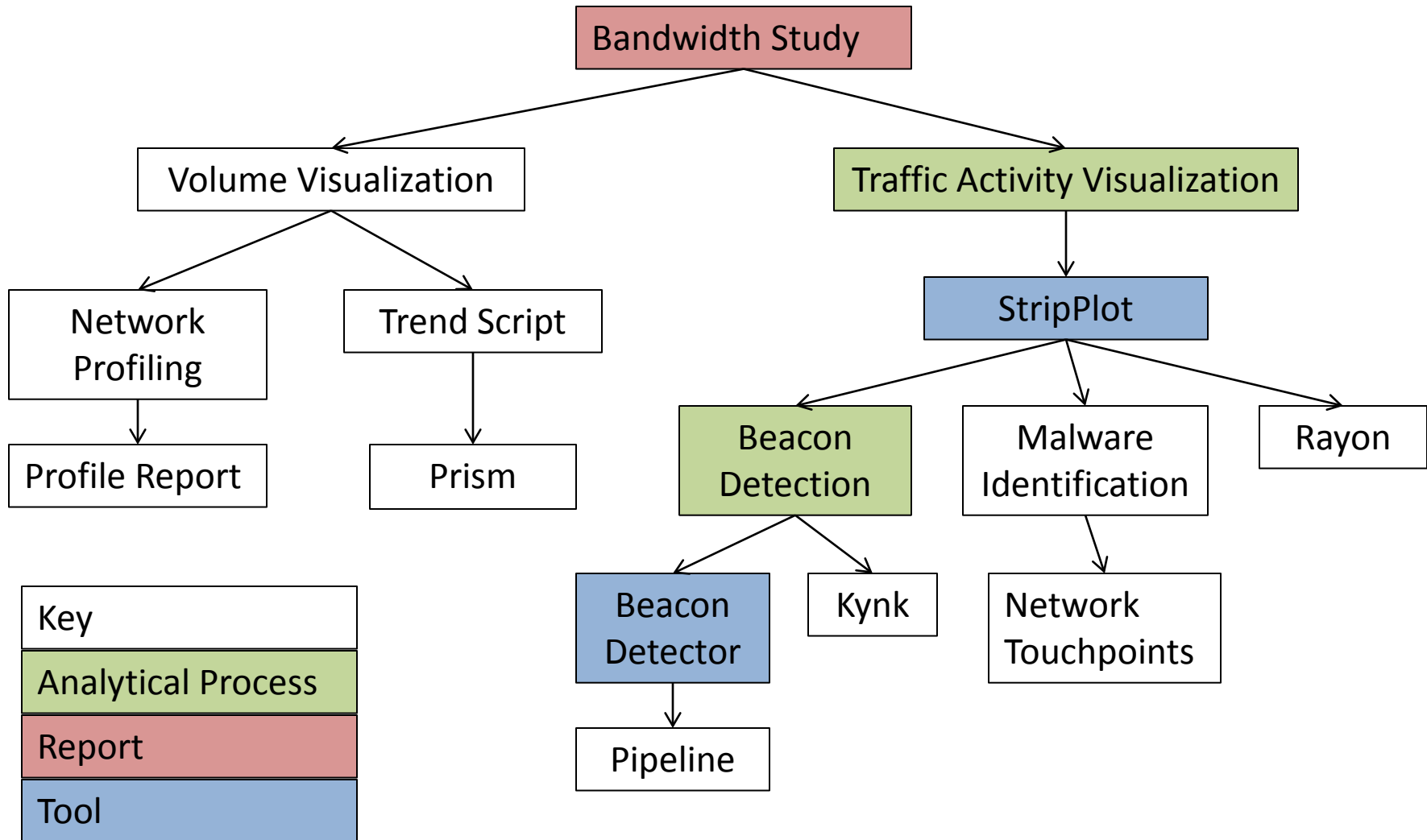
So... if we can find beacons we can find botnets, Right?

Yes, if you can distinguish a beacon from other regular behavior

Which is hard



Overview



Paul's Beacon Detector

Beacons exhibit regular behavior

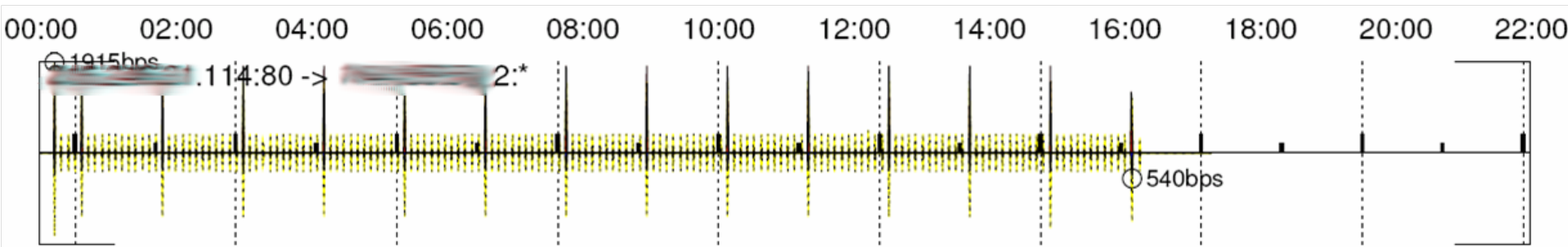
- A series of connections or connection attempts
- Between the same two IP addresses
- At regular time intervals

Implemented a Finite State Machine to find

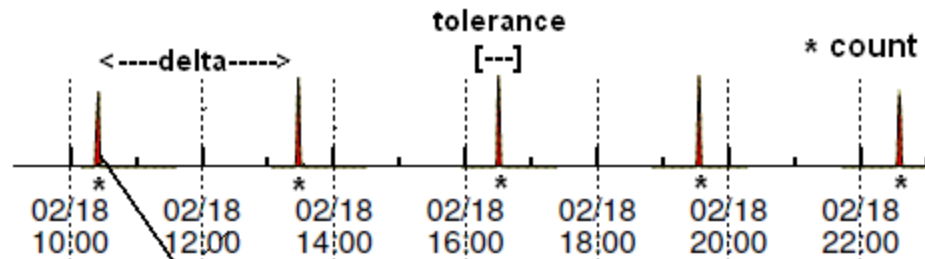
- X or more flows (5 flows)
- At regular interval of Y ($Y \geq 5$ minutes)
- With a tolerance of Z percent (5%)



Beacon Detection



Characterizing Beaconsing Activity



Traffic characteristics

- protocol
- TCP flags
- bytes
- duration



Did it work?

Did it find regular behavior?

- Yes, rather a lot of it

Did it find botnet beacons?

- Probably but hard to distinguish from all the other stuff

What other stuff?

- NTP, News updates, email updates, DNS...



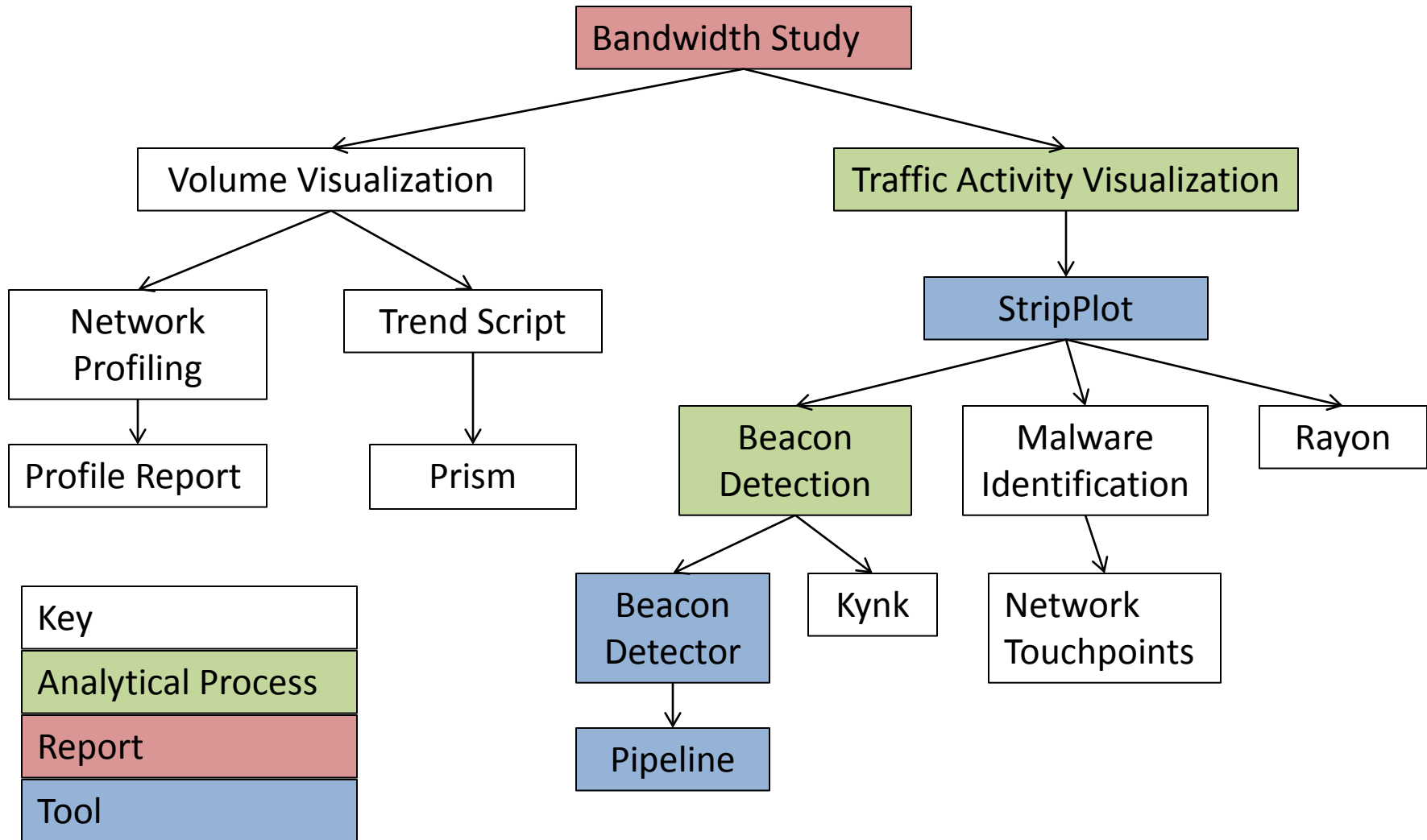
Can it be made better?

Three ways that we know of

- Find more regular behavior
 - Missing flows
- Additional information
 - Actual botnet beacon characteristics
 - Any other information that can be used with flow analysis
- Extreme whitelisting
 - Keep track of everything that beacons, and ignore it
 - Only look for new stuff
 - Keep track of the beaconing addresses for the last 30 days
 - Whitelist them



Overview



Get results sooner

Traditional SiLK commands find flows in the repository

To get the most recent, set the search time and run it in cron, but how often

- Run cron too often and one doesn't finish before the next one starts
- Run it less often and you wait longer than necessary

We want to look at flows as soon as they are available



Pipeline fills that role

Pipeline runs continuously and processes SiLK files as they are written

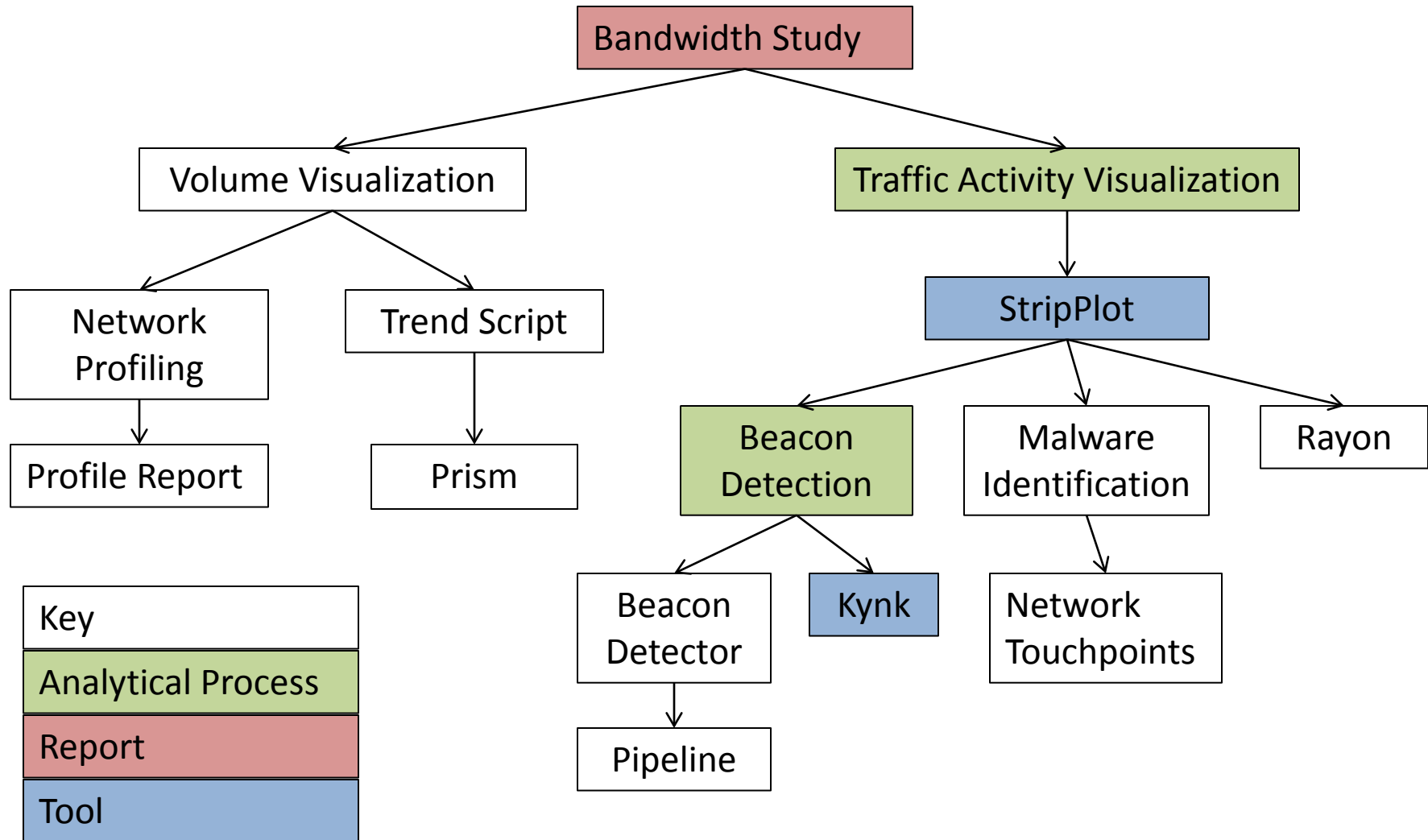
Pipeline has its own unique filtering strategy

Paul's Finite State Machine was implemented in Pipeline

It will alert as beacons (instances of regular behavior) are found



Overview



Eight Different Beacon Detectors?

Motivation

- Beacon detection is either very useful or a very shiny object: I know of at least 8 implementations, 9 if you count stripplot.
- Saw beaconing in strip plots of RAT
- Recognized utility of finding beacons to detect certain RATs
- Concluded that "eye charts don't scale"
- Determined to explore algorithmic approaches



YABD_[1] – Yet Another Beacon Detector?

Activities

- Explored different algorithms, implemented several
- Performed analysis of running time
- Identified common sources of false positives
- Generated RAT traffic in lab for testing
- Explored live data

[1] Biologists use YABD as an index of the health of deer in relation to carrying capacity.



From Eye Charts to...

Outcomes

- Two first generation beacon detectors
- One second generation detector in pipeline
- Tools delivered to different analyst communities with mixed levels of adoption.



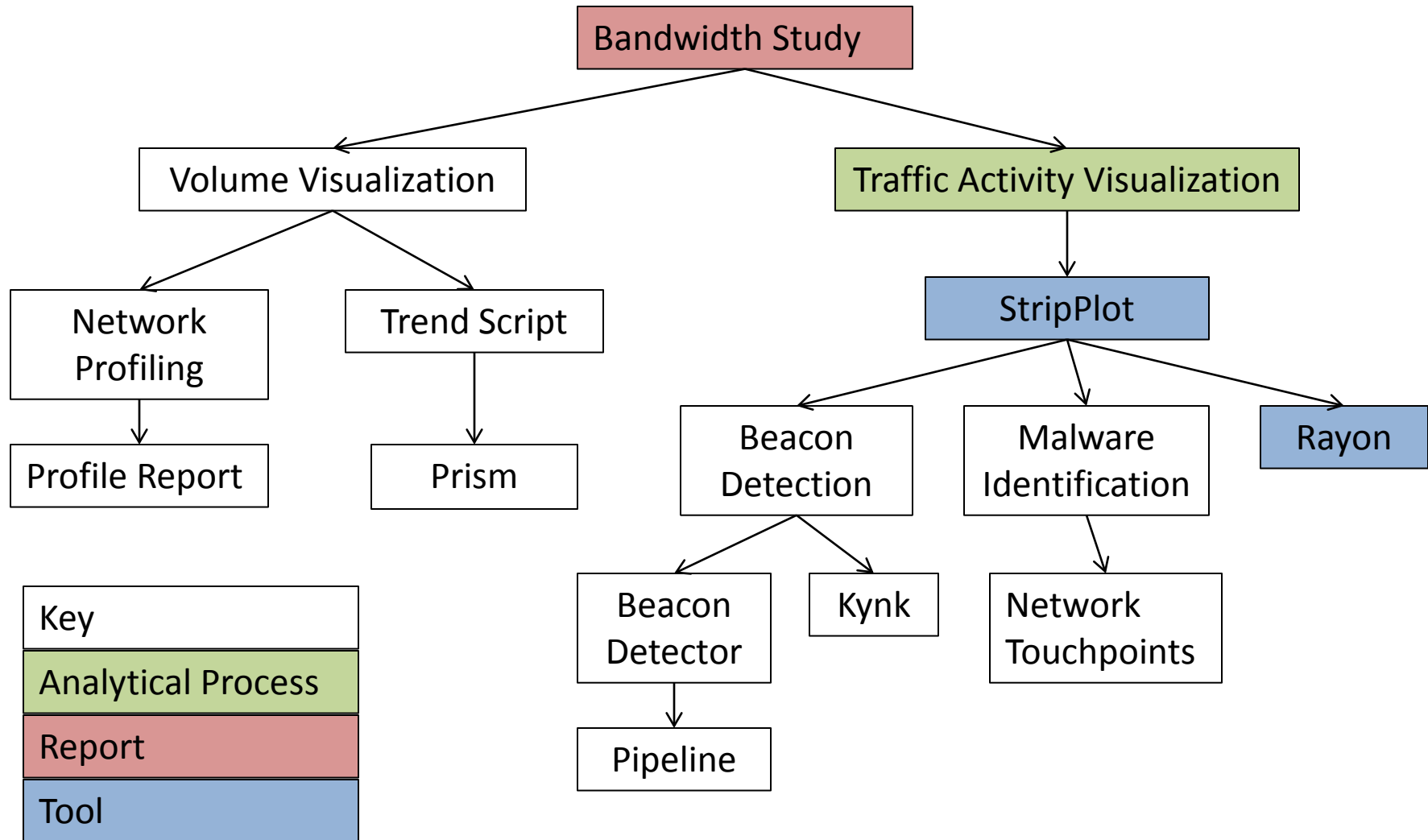


Lessons Learned

- Your first thought on algorithms may not be right
- You need need a large sample of ground truth to test against
- Algorithms that work on a few samples may not work in the wild.
- It's hard to generate realistic background data.
- False positives are common.
- Need to socialize more with analyst community.
- Adoption is tied to perceived utility of the tool, ownership the analysts feel of it (homegrown tools win), and their trust in the person/organization providing the tool to meet their specific needs.



Overview



The Rayon Viz Library

Several analytics had visualization requirements in common

StripPlot pushed Gnuplot to its limits

It was time to move away from “Analyst Code”



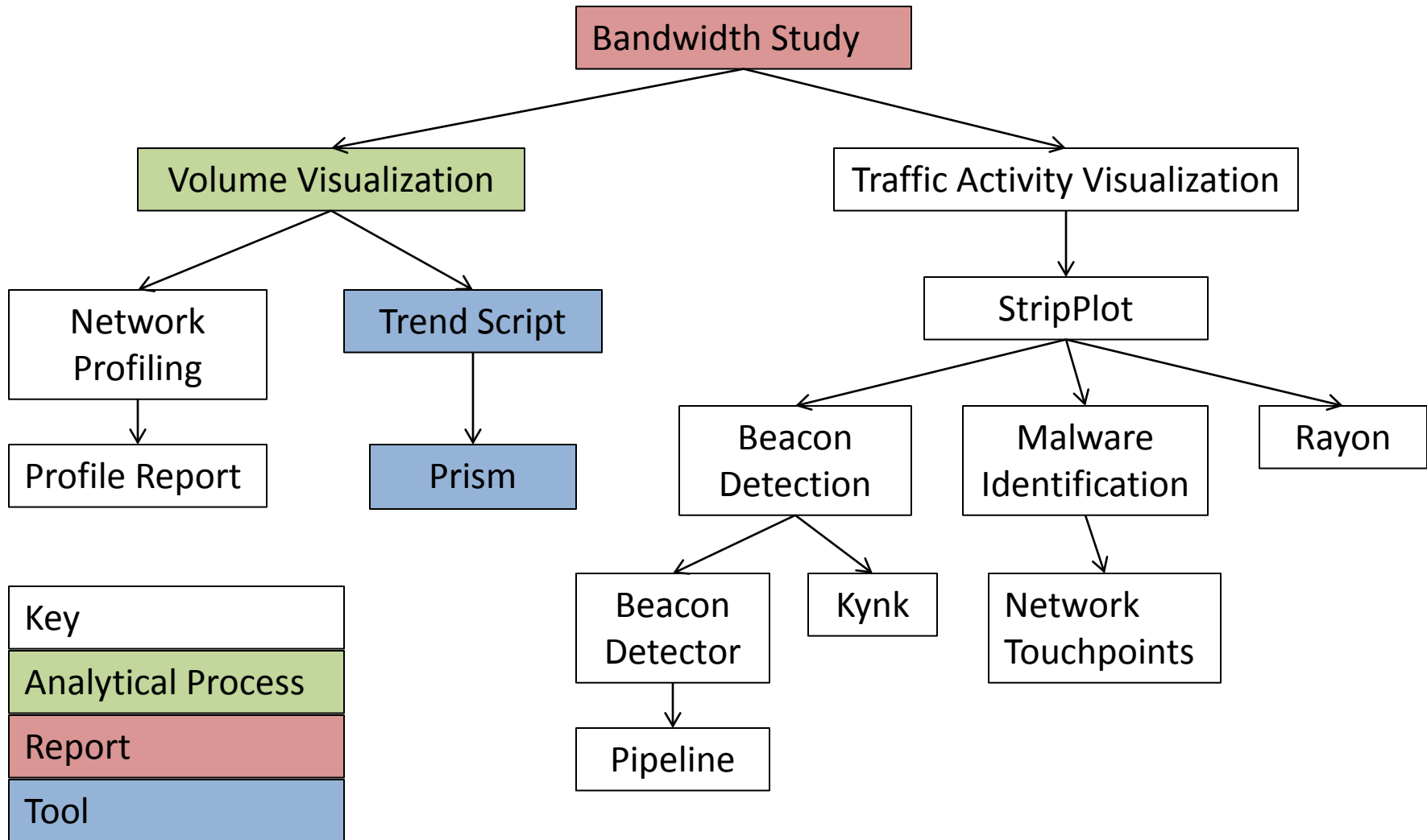
Why didn't he call it Yet Another Graphics Package

Phil Groce of the NetSA Development team

- gathered requirements
- wrote a set of “flow aware” graphics primitives
- wrote several applications using the primitives
- released it to the world as Rayon
 - <http://tools.netsa.cert.org/rayon/index.html>
- ask us later if you don't get the play on words



Overview



Prism

There was a renewed interest in Trend Script

But it is an analyst's tool for specific tasks

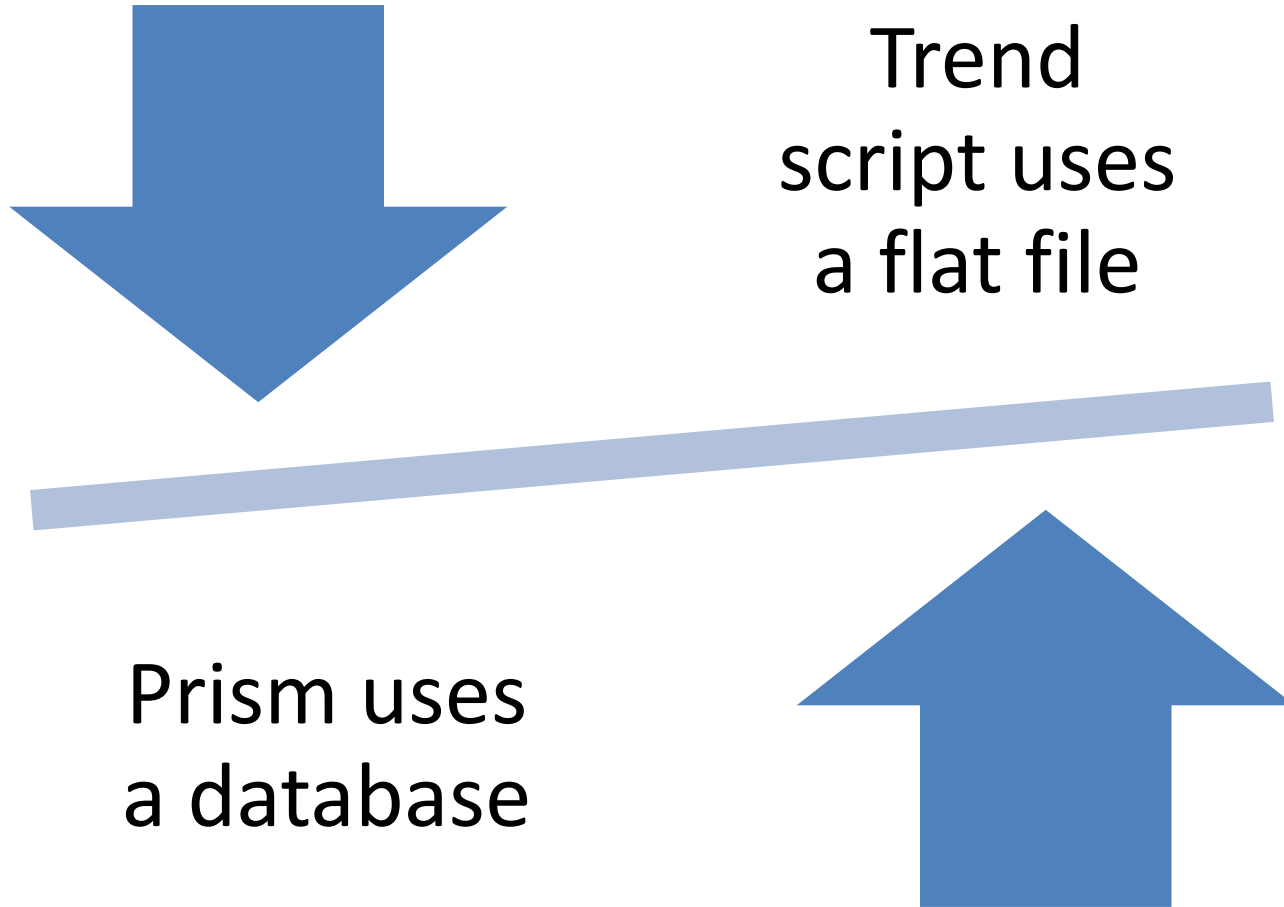
A continuous volume display has other requirements

Prism is a re-write of the Trend Script by NetSA

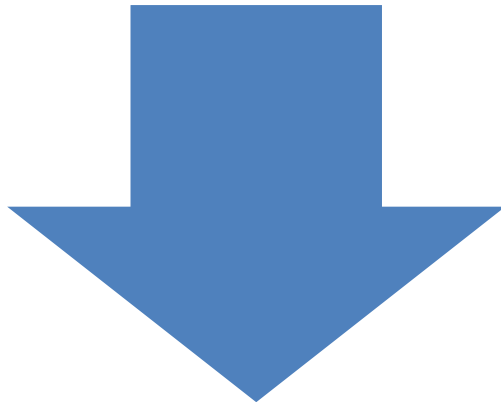
Development Team member John Prevost



Prism vs Trend Script



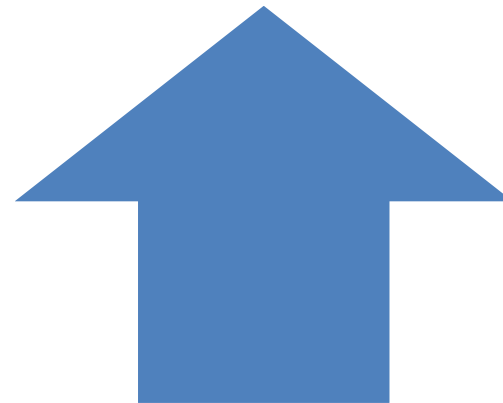
Prism vs Trend Script



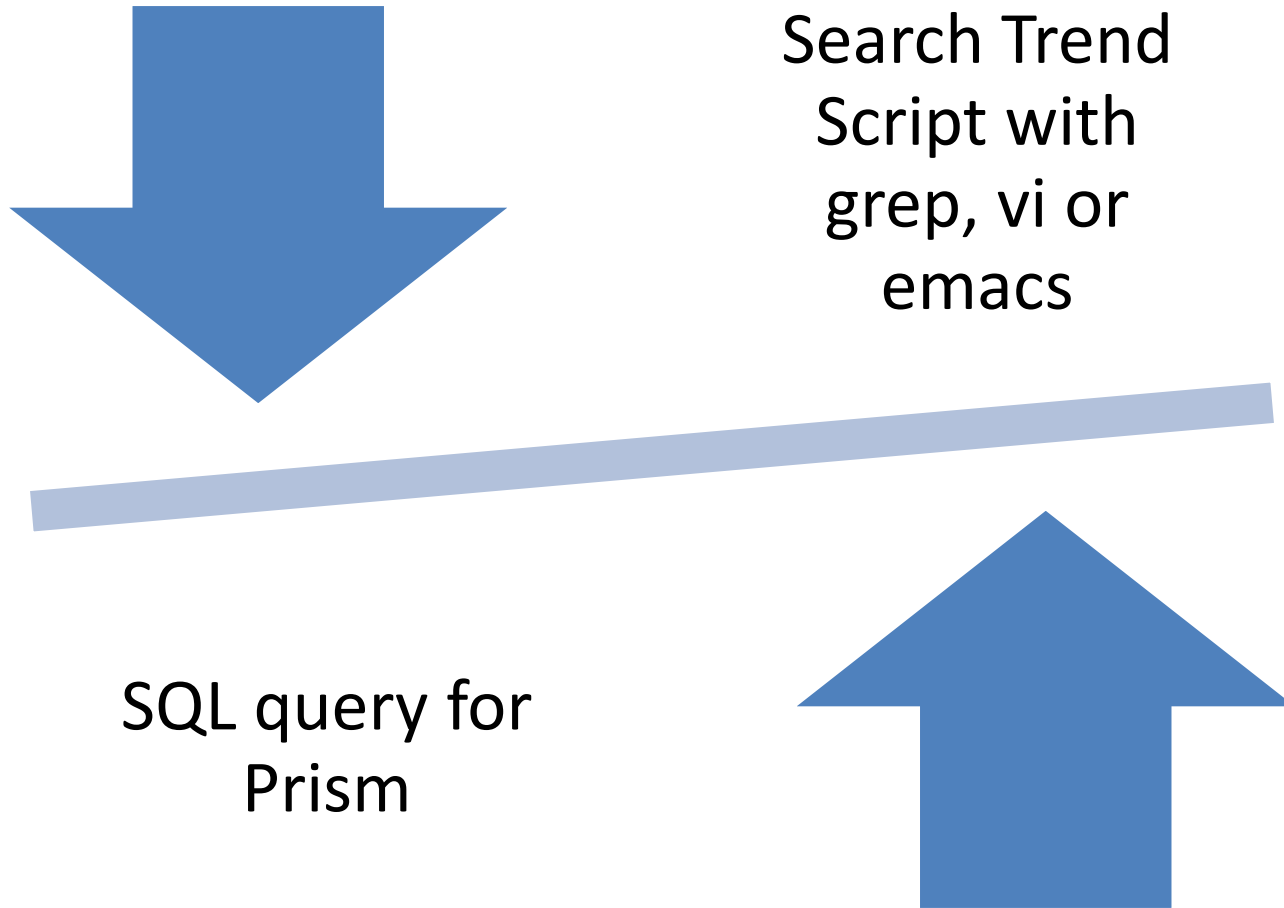
One offs in
Trend Script
are easy



There is no
such thing
in Prism



Prism vs Trend Script



Conclusions

One thing leads to another

“If we knew what we were doing, it wouldn’t be called research, would it?” A. Einstein

Don’t be afraid to scrap something and start over





Paul Krystosek `pnk@cert.org`

George Jones `gmj@cert.org`

Sid Faber `sfaber@cert.org`

Network Situational Awareness Group

CERT Software Engineering Institute

Carnegie Mellon University

Pittsburgh, PA

