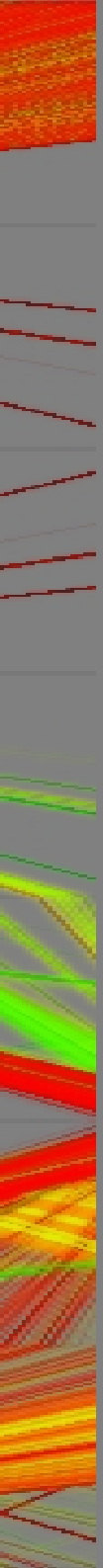


Picviz Workshop

iAWACS 2009

Sebastien Tricaud

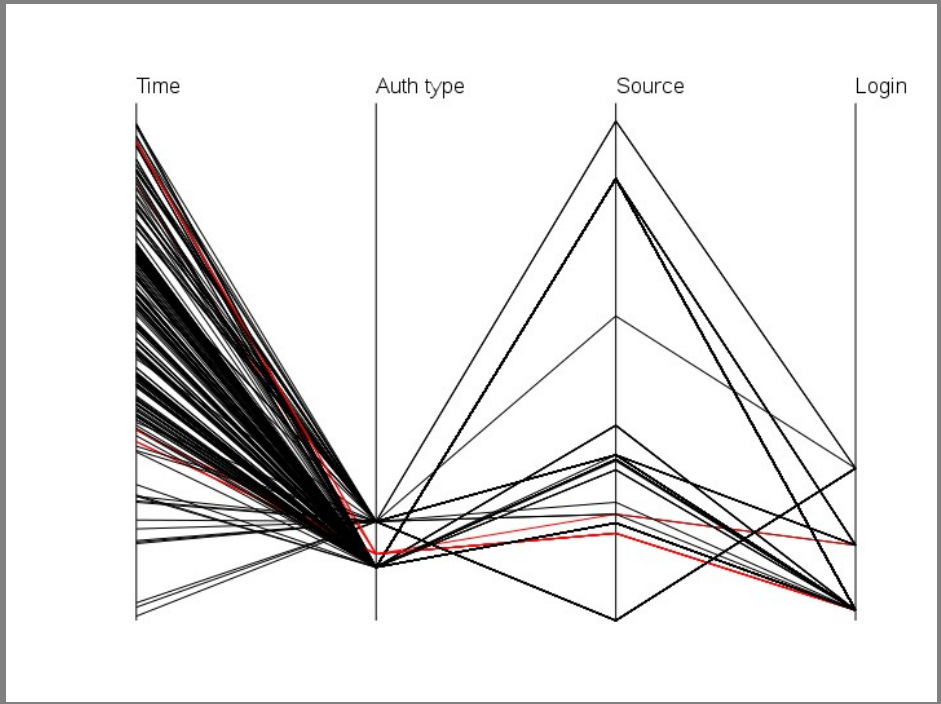
Summary

- Why Picviz?
- Architecture
- The PGDL Language
- Generating images
- Analyzing authentication logs
- Analyzing Apache logs
- Time for a coffee

Why Picviz: history

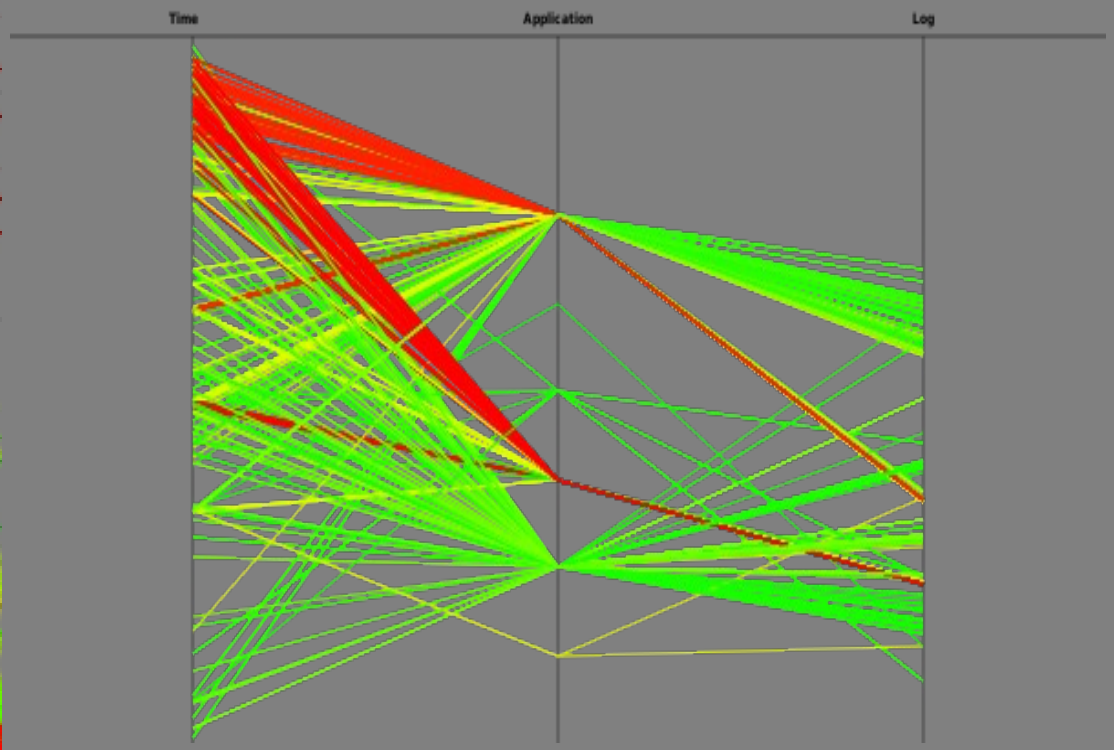
- Created in summer 2008 by Sebastien Tricaud and Philippe Saade
- One problem: how to handle your IDS logs at a national scale?
- One answer: using visualization
- Which visualization technique can deal with an infinity of events and multiple dimensions?
- => Parallel coordinates
- We can also extract relevant mathematical properties from PC!

What are Parallel Coordinates?

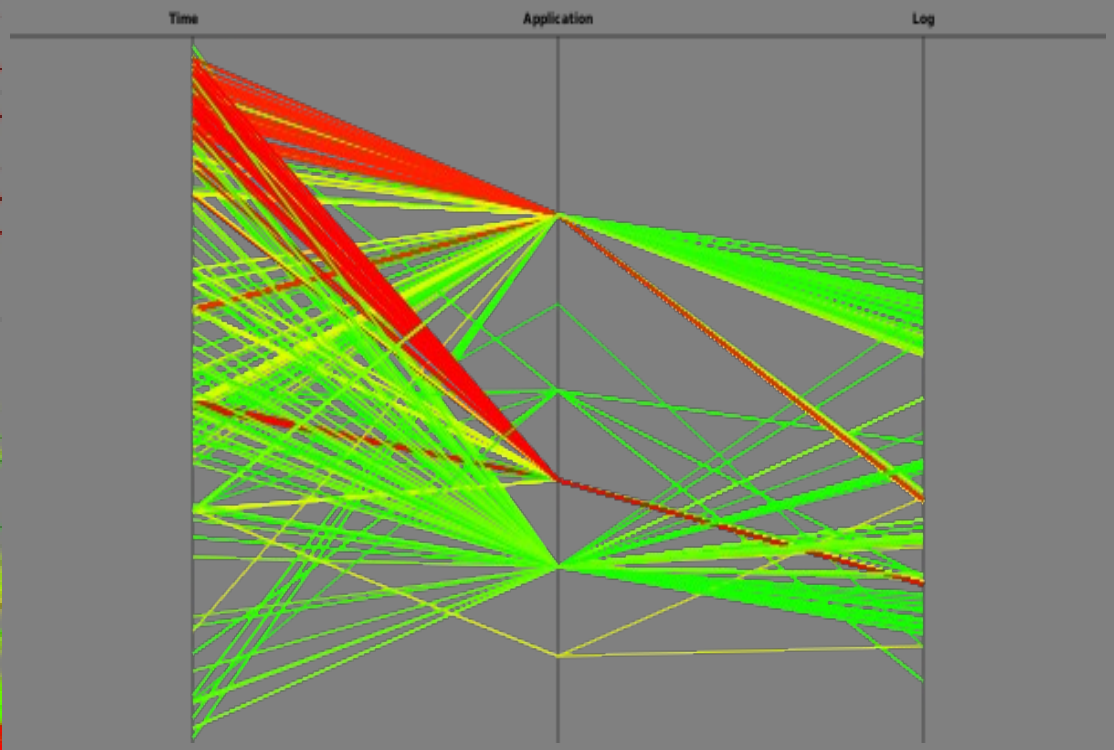


- A visualization technique invented by Maurice d'Ocagne and rediscovered by Alfred Inselberg in the 70s
- One axis = one dimension
- A plot = the value on the axis

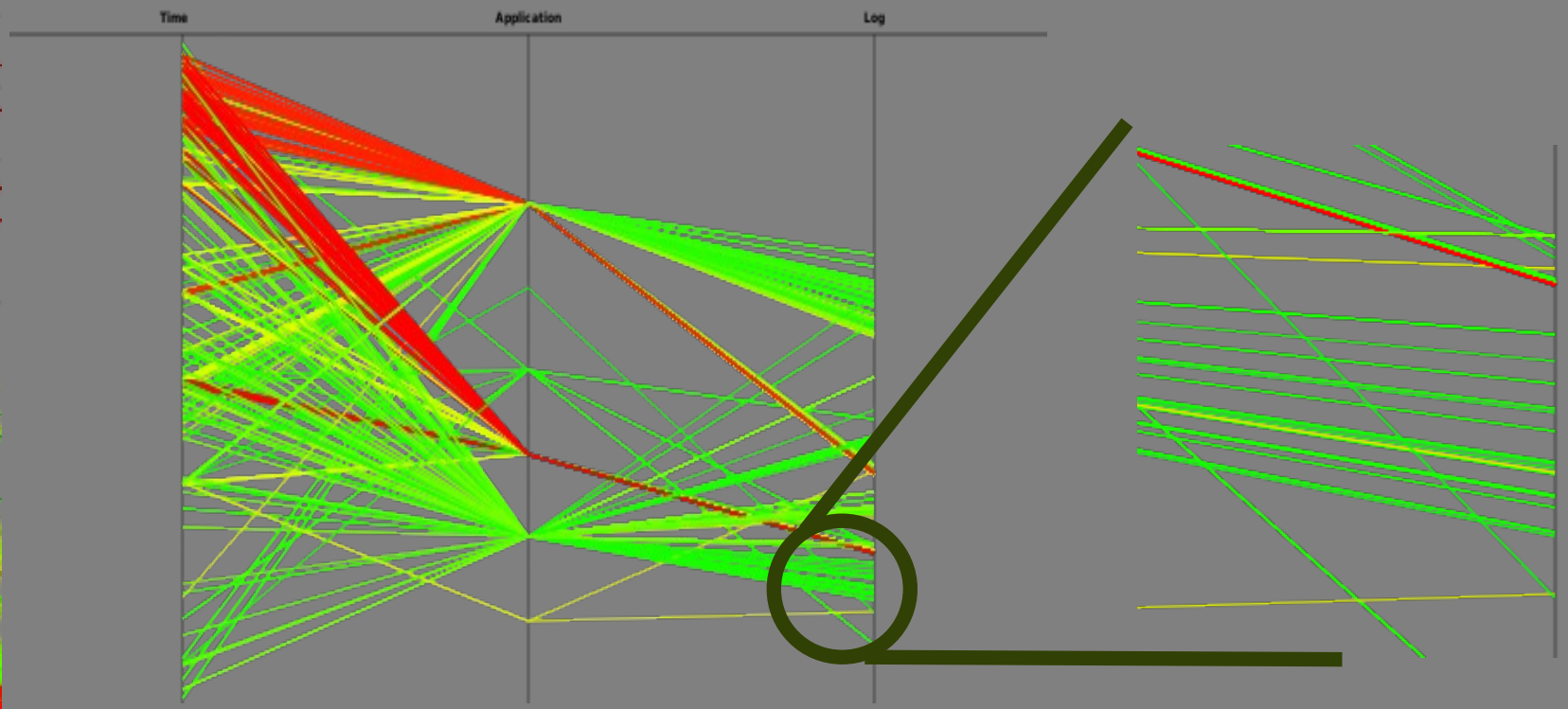
Attack seen by Picviz



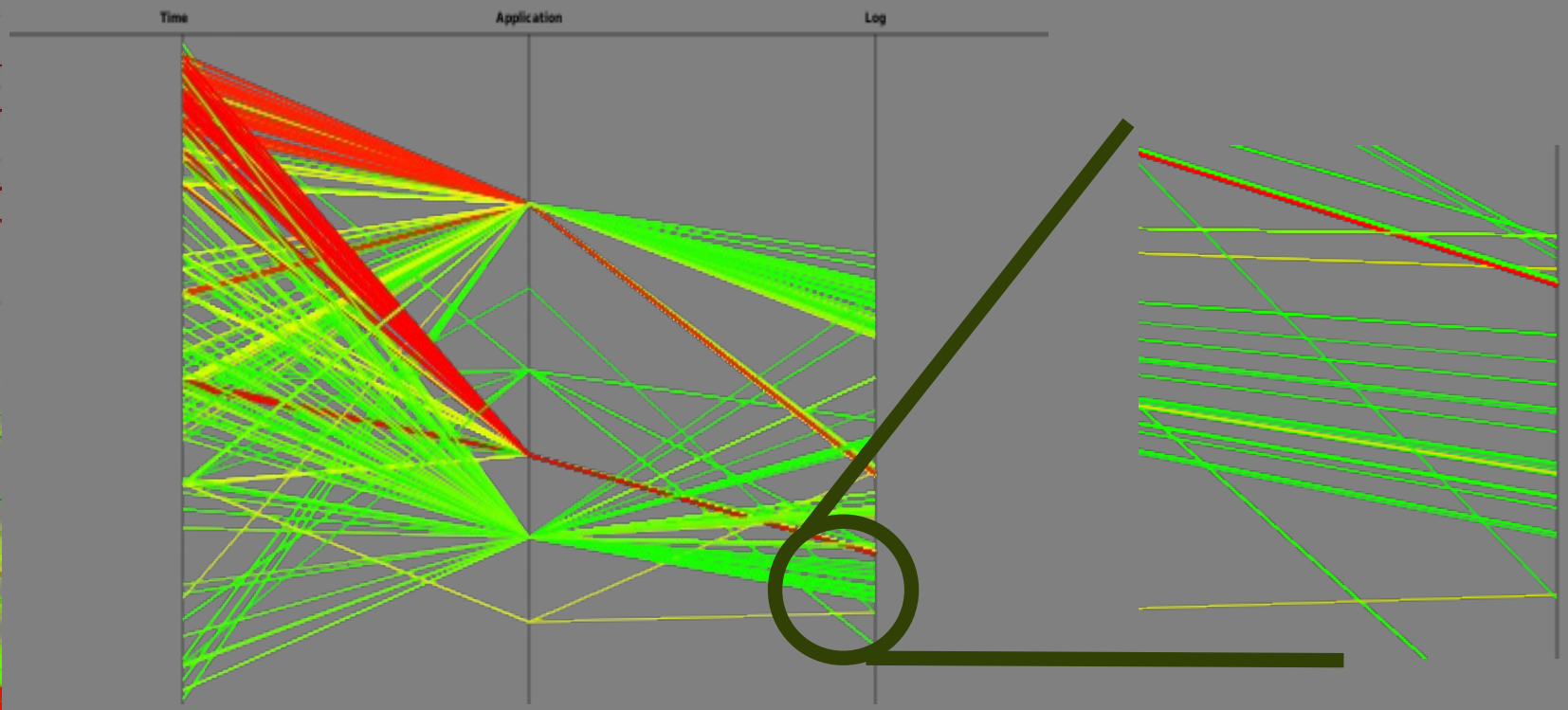
Attack seen by Picviz



Attack seen by Picviz

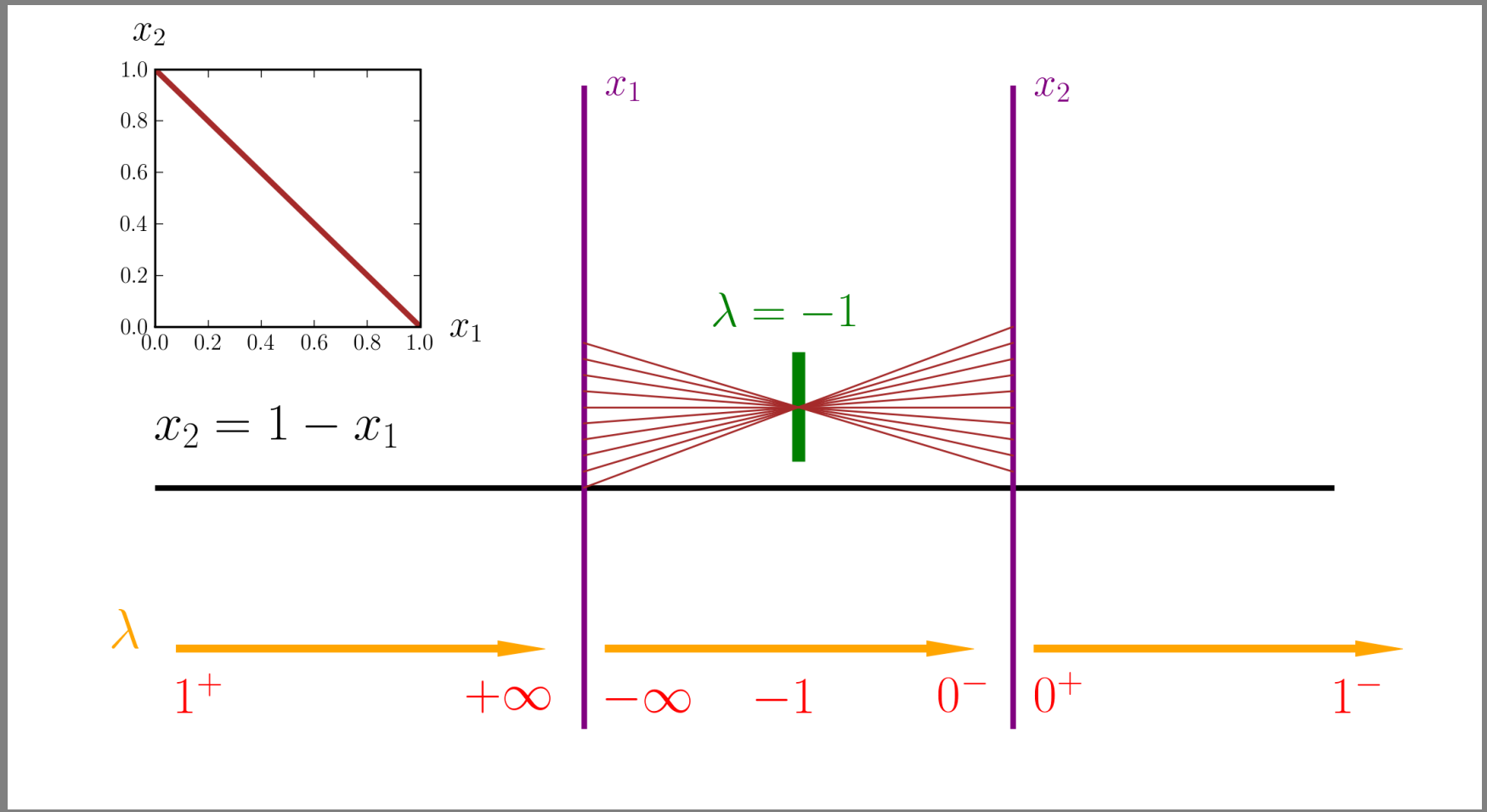


Attack seen by Picviz

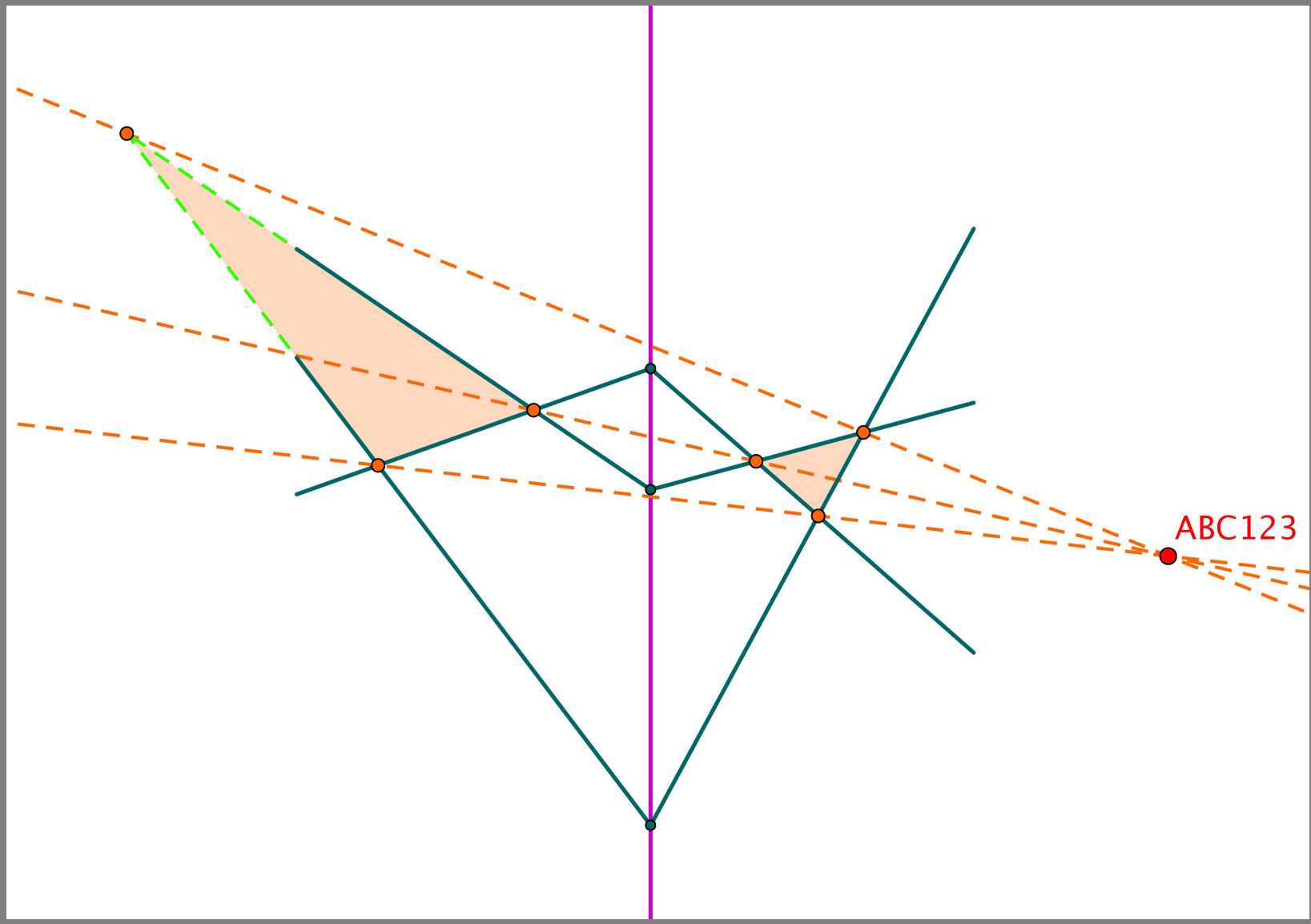



```
May  1 05:42:01 fw1 sshd[31911]: Invalid user test from 10.3.12.3
May  1 05:42:01 fw1 sshd[31957]: Invalid user john from 10.3.12.3
May  1 05:42:01 fw1 sshd[31957]: Invalid user admin from 10.3.12.3
May  1 05:42:01 fw1 sshd[31957]: Invalid user svn from 10.3.12.3
May  1 05:42:01 fw1 sshd[31957]: Invalid user printer from 10.3.12.3
```

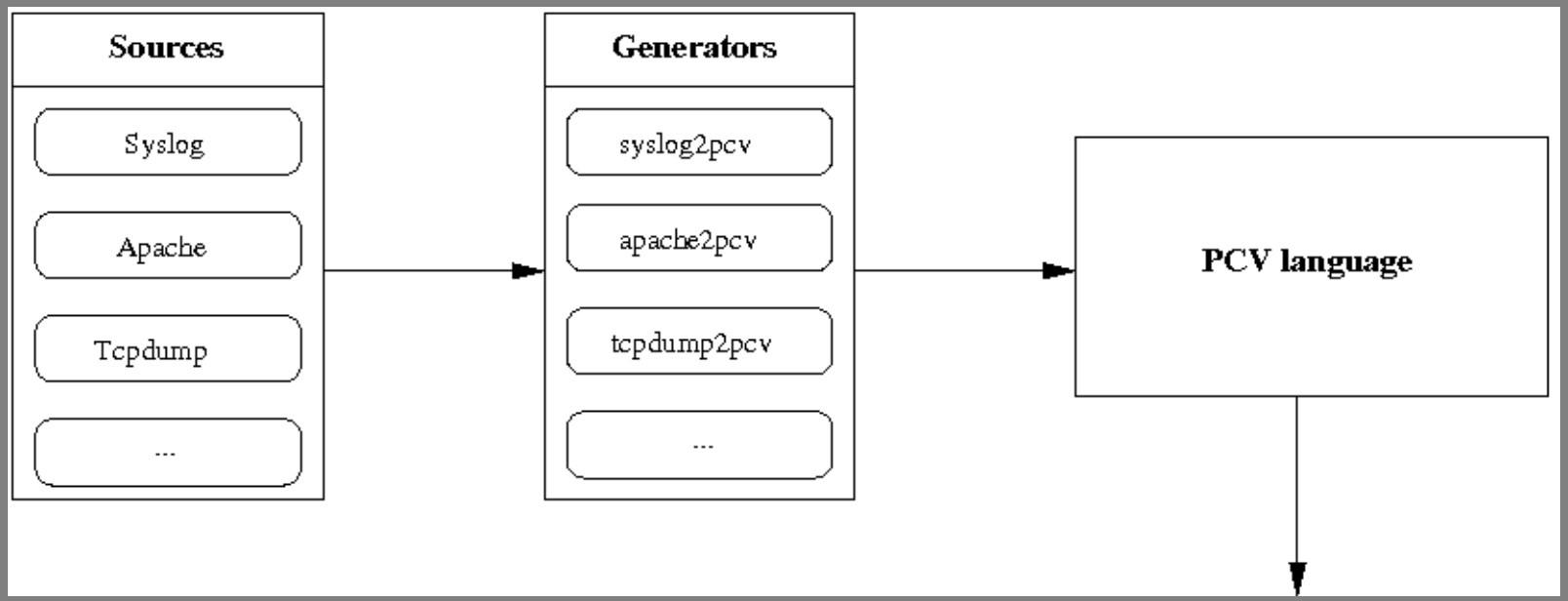
Parallel Coordinates & Maths



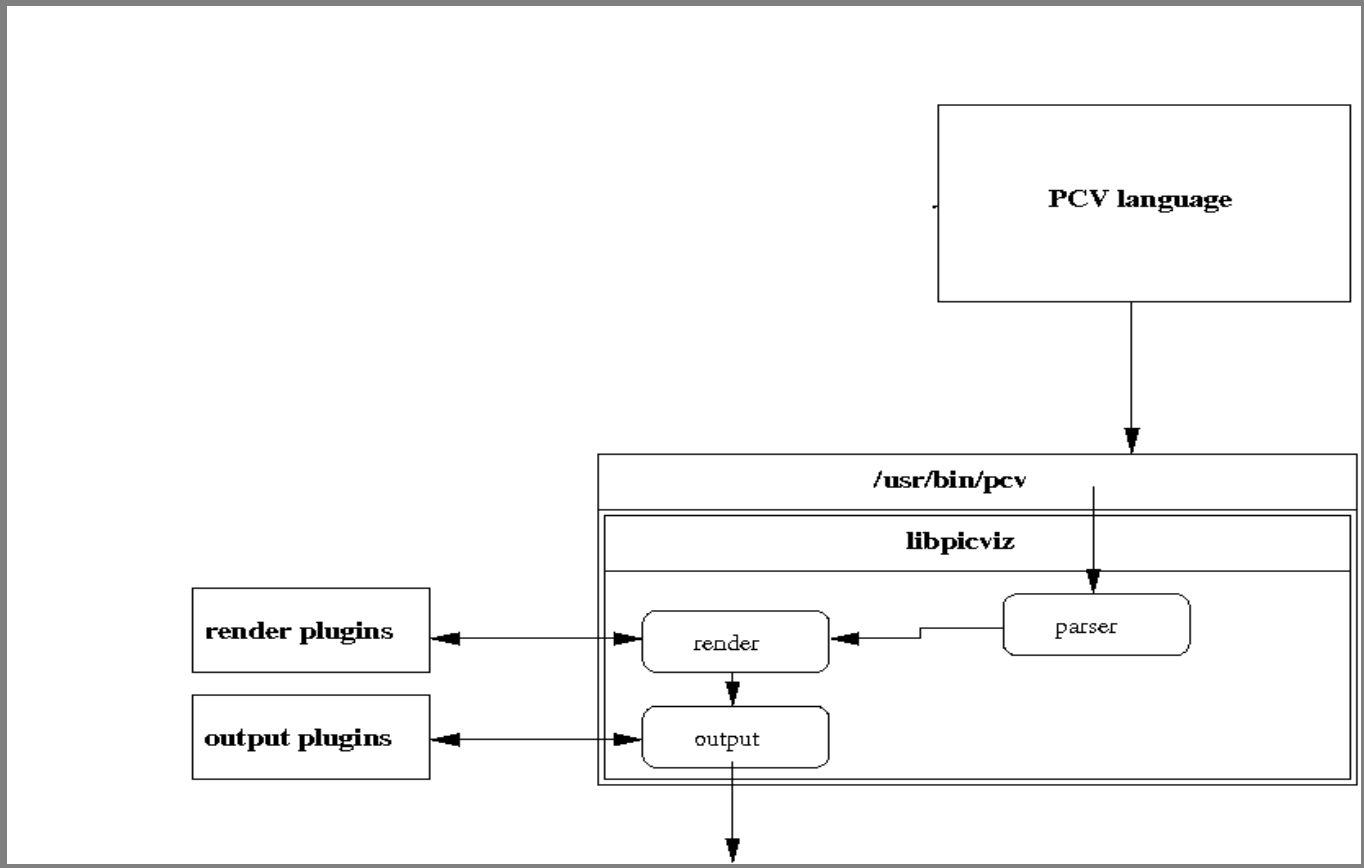
Parallel Coordinates & Maths



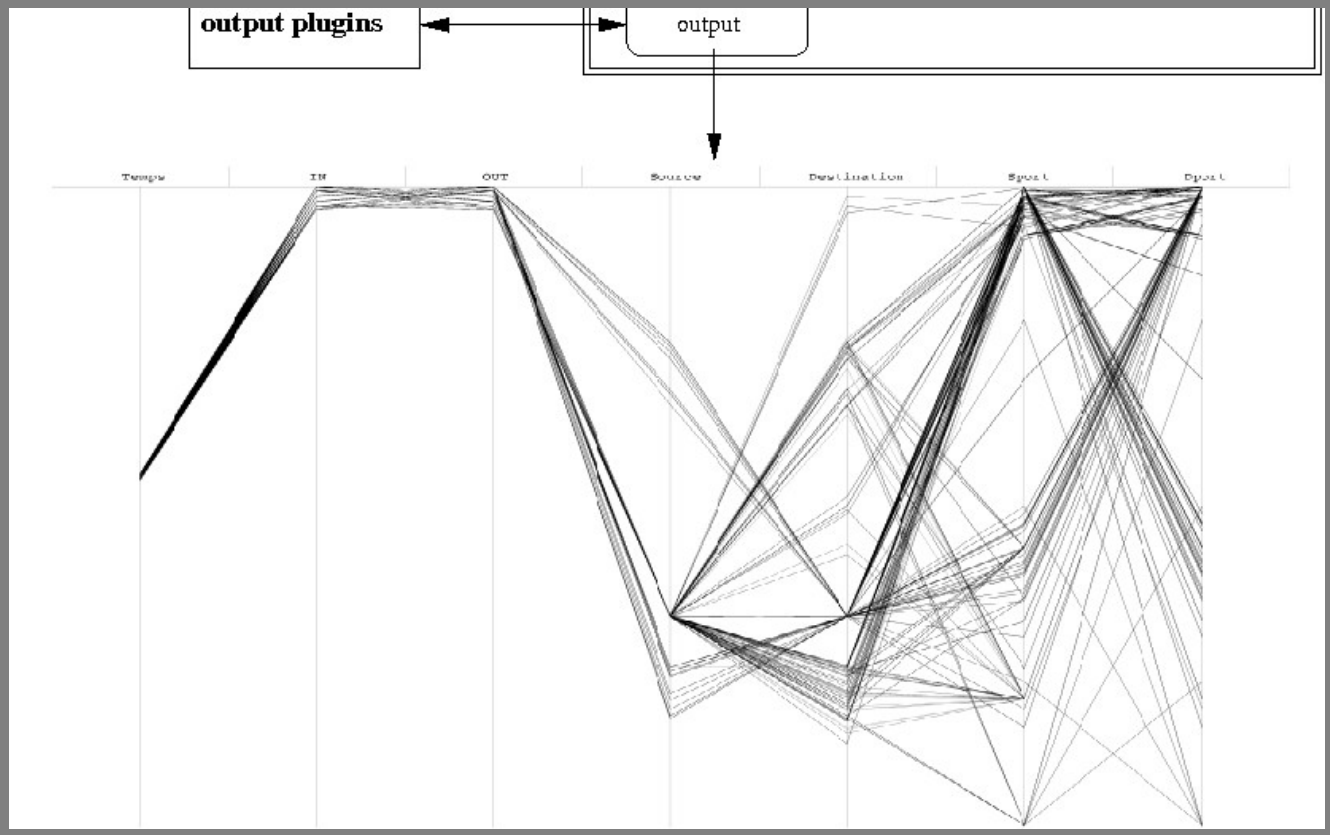
Picviz architecture



Picviz architecture



Picviz architecture



What we need to do

- Just like Graphviz, Picviz has its own language
- It is easy to take CSV files, CSV are not rich enough though
- Write the axes, their properties and add the data

```
axes {
    timeline axis1 [label='Time'];
    enum axis2 [label='User'];
}
data {
   axis1='19:42', axis2='root' [color='red'];
}
```

We have several variables

- enum, integer, ipv4, port, string, ...
- In the axis section, this is « variable axis »
- Variable can be replaced by a number « 500 axis », 500 will be the maximum this axis can take
- ...

Helper to write a parser: syslog parser

- Python class PicvizLogParser

```
import PicvizLogParser

axes = ['time', 'machine', 'application', 'log']

pp = PicvizLogParser.PicvizLogParser(sys.argv[1], axes,
header="Syslog picviz analysis")


pp.axesdict['time']['type'] = 'timeline'

pp.axesdict['log']['type'] = 'string'


pp.setPCRE(r'\w+ ?\d+ (\d+:\d+):\d+ (\w+) (\w+)\S+ (.*)')

pp.run(pp.defaultprintcb)
```

Creating the image

```
$ pcv -Tpngcairo syslog.pgdl -o syslog.png
```

```
$ pcv -Tpngcairo syslog.pgdl -o syslog.png -Acurves
```

```
$ pcv -Tpngcairo -rr -a syslog.pgdl -o syslog.png
```

Frequency analysis

$ pcv -Tpngcairo **-Rheatline** syslog.pgdl -o syslog.png

$ pcv -Tpngcairo **-Rheatline -Avirus** syslog.pgdl -o syslog.png

Filtering

[value|plot|freq] relation value [{on axis n}|{on axes}] {[and|or]} ...

$ pcv -Tpngcairo file.pgdl -o file.png 'value = "foobar" on axis 3'

$ pcv -Tpngcairo file.pgdl -o file.png -Wpcre 'value = ".*[Ff]oobar.*" on axis 3'

$ pcv -Tpngcairo file.pgdl -o file.png 'plot > 100" on axis 3'

$ pcv -Tpngcairo file.pgdl -o file.png 'plot <= 10%" on axis 2 and plot > 50% on axis 3'

$ pcv -Tpngcairo file.pgdl -o file.png -Rheatline 'freq < 0.002'

Data Analysis

- On your USB key, compare the different scanners
- Download Apache access logs, analyze them: find evil requests
- Download authentication logs, someone breaked into it. Find it!

Future

- http://www.wallinfire.net/picviz
- sebastien@honeynet.org
- Thank you!