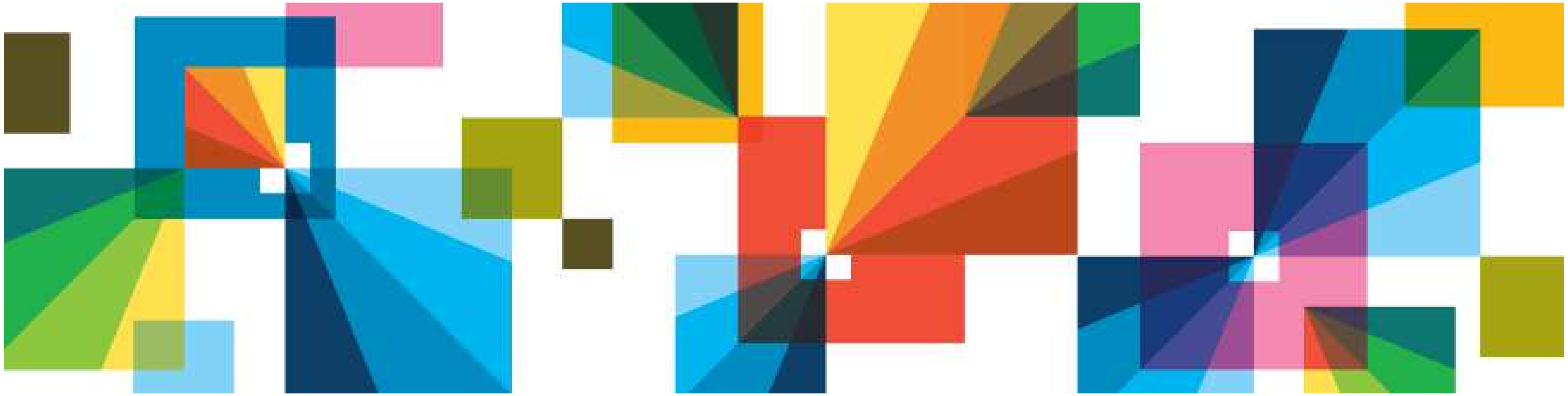
Yifat Yulevich, Certified Senior IT Architect
Alex Pyasik, Software Engineer
Leonid Gorelik, IT Specialist

Streams for Real-Time Analytics

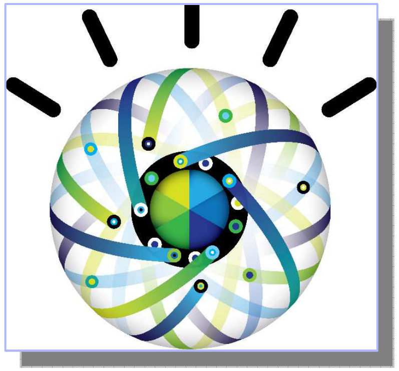


InfoSphere Streams Overview

Real Time Security Analytics

DNS Cache Poisoning Demo

Real-Time Intelligence Generation



InfoSphere Streams Overview

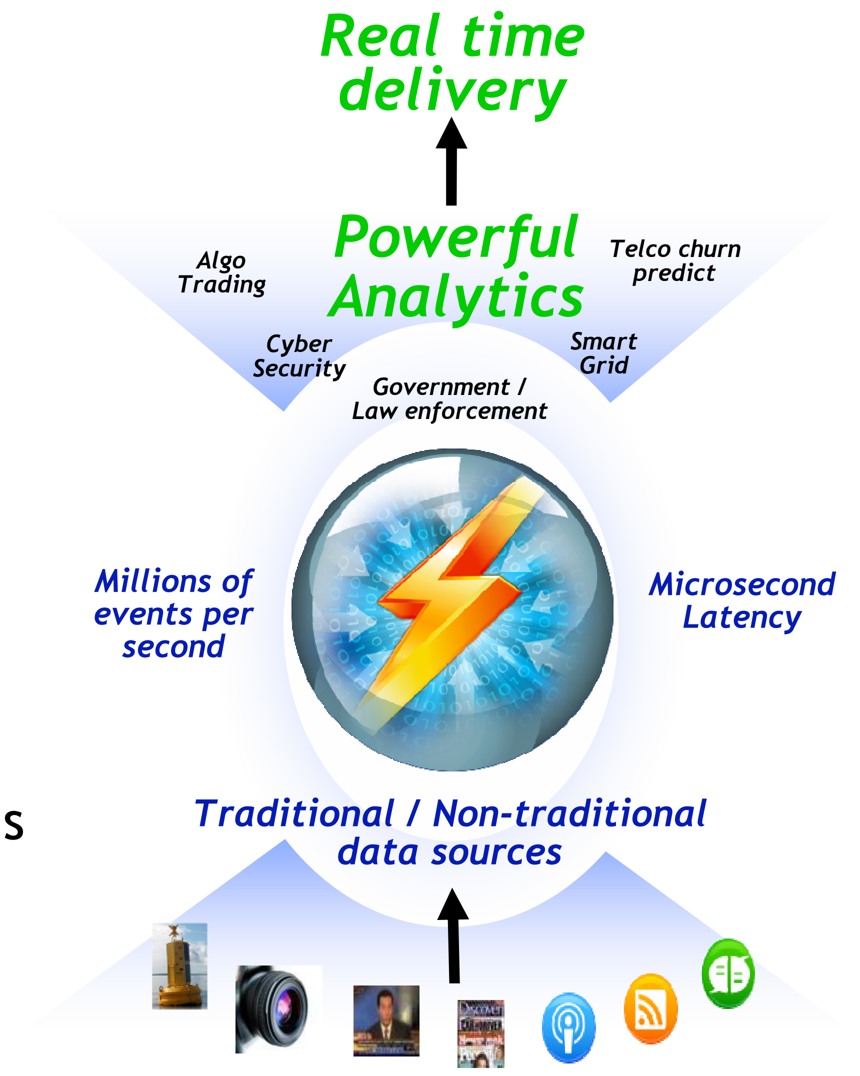
A Platform to Run In-Motion Analytics on **BIG** Data

Volume Petabytes per day

Variety All kinds of data

All kinds of analytics

Velocity Insights in microseconds



Real time delivery

Powerful Analytics

Algo Trading

Telco churn predict

Cyber Security

Smart Grid

Government / Law enforcement

Millions of events per second

Microsecond Latency

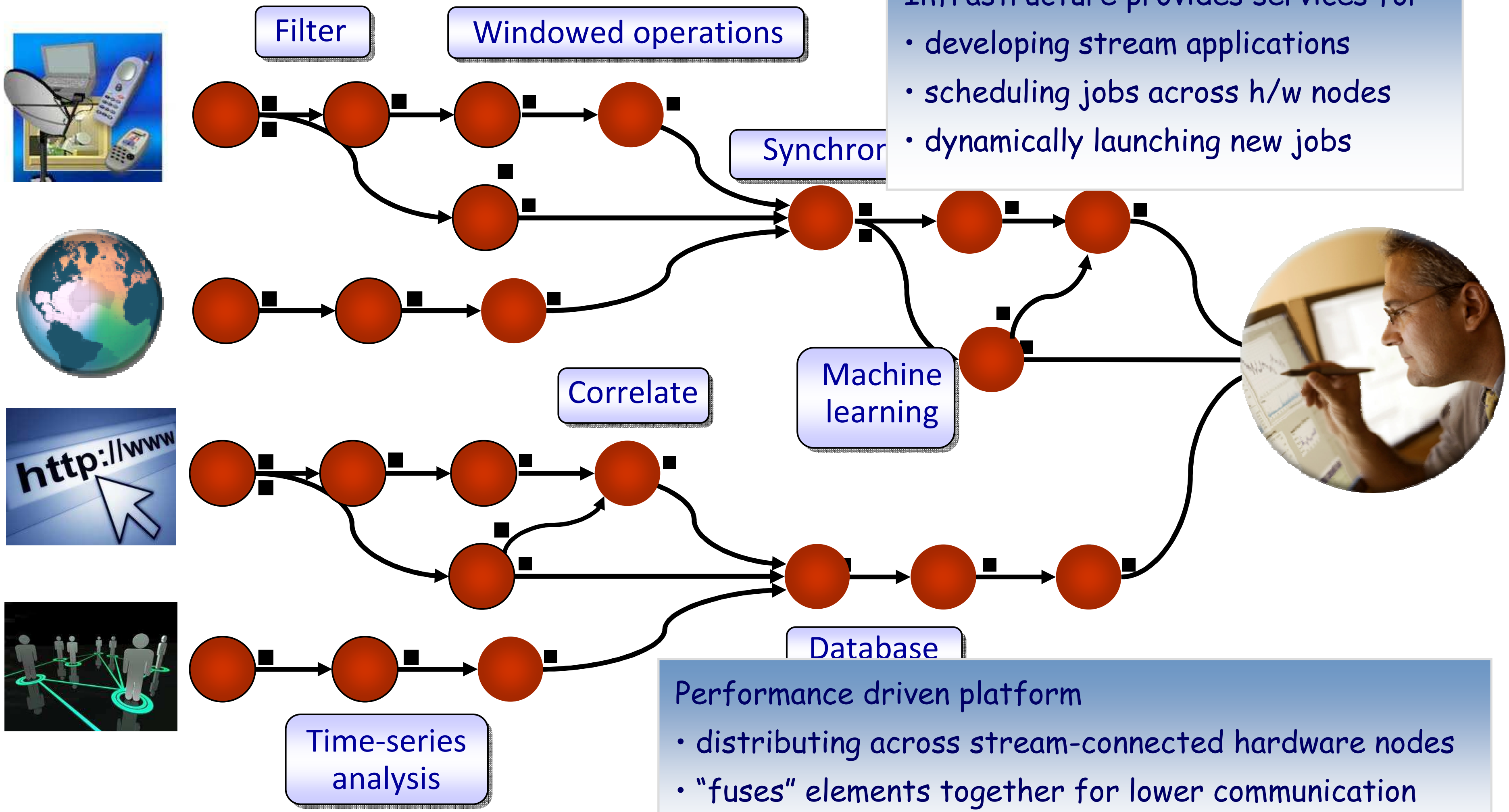
Traditional / Non-traditional data sources

- Continuous ingestion
- Continuous analysis

puting Paradigm

Infrastructure provides services for

- developing stream applications
- scheduling jobs across h/w nodes
- dynamically launching new jobs

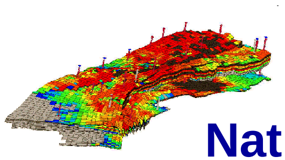


Performance driven platform

- distributing across stream-connected hardware nodes
- “fuses” elements together for lower communication latency

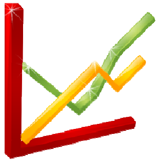
n

InfoSphere Streams in various industries



Natural Systems

- Seismic monitoring
- Wildfire management
- Water management



Stock market

- Impact of weather on securities prices
- Analyze market data at ultra-low latencies

Radio Astronomy

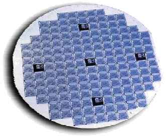
- Detection of transient events

Transportation

- Intelligent traffic management

Telecommunications

- Processing of CDRs for Business Intelligence, Revenue
- Assurance, etc.



Manufacturing

- Process control for microchip fabrication

Cyber Security

- Real-time network monitoring

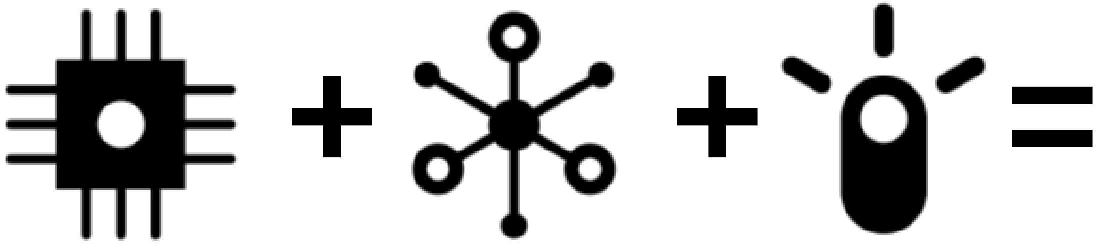
Health & Life Sciences

- Neonatal ICU monitoring
- Epidemic early warning system
- Remote healthcare monitoring

Real-Time Analytics **for Cyber Security**

The Opportunity of a Smarter Planet

Every natural system and man-made system
is becoming interconnected, instrumented and intelligent



Smarter Utilities

Smarter Cities

Smarter Food

Smarter Transportation

Smarter Oil & Gas

Smarter Public Safety

Traditional Attack

- ⌘ Discover the attack
- ⌘ Investigation, evidences collection
- ⌘ Detect and analyze the attack patterns
- ⌘ Create signatures
- ⌘ Apply them in the appropriate systems

Machine Learning

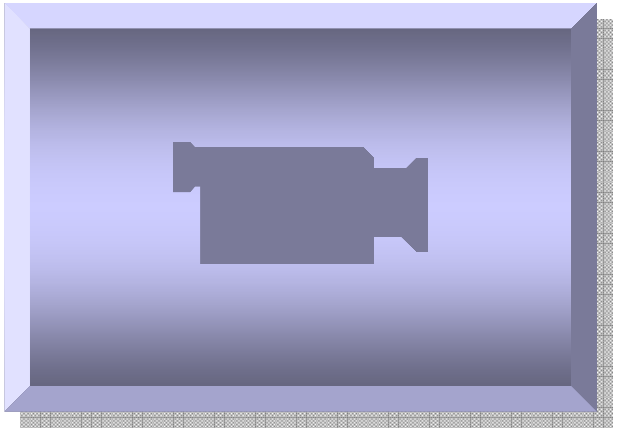
Anomaly Detection Concept

- Anomaly detection - finding patterns in data that do not conform to expected behavior.

- By observing various data sets and activities, the anomaly detection systems can classify the behavior and determine if it is either normal or anomalous.

- Unlike signature-based cyber security systems, which can only detect attacks for which a signature has previously been created, anomaly detection is based on behavioral patterns, heuristics and rules and will detect behavior that falls outside of normal system operation.

Learning Algorithms and Anomaly Detection

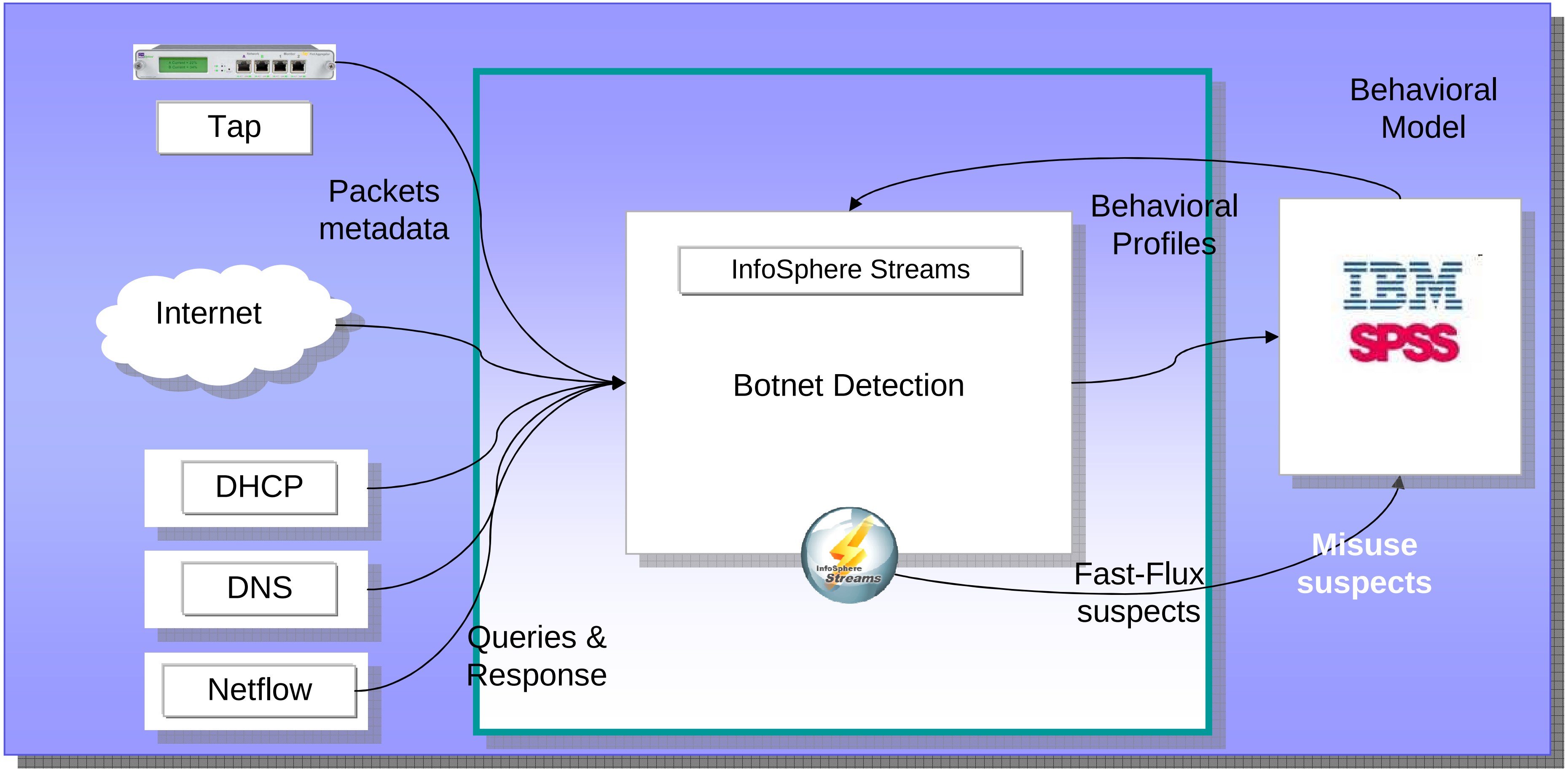


Shallow vs Deep Content Inspection

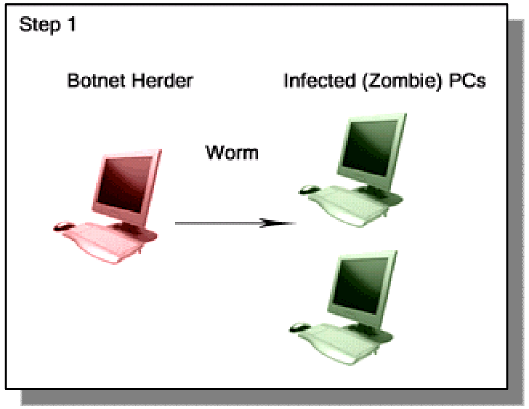
Streaming analytics provides a broad spectrum of analyses including

- advanced behavioral analytics (such as per-host / per-user / per-network-entity level)
- deep content inspection
- alert fusion and correlation
- anomaly detection
- machine learning based techniques

Solution Outline



Definition: What are botnets?



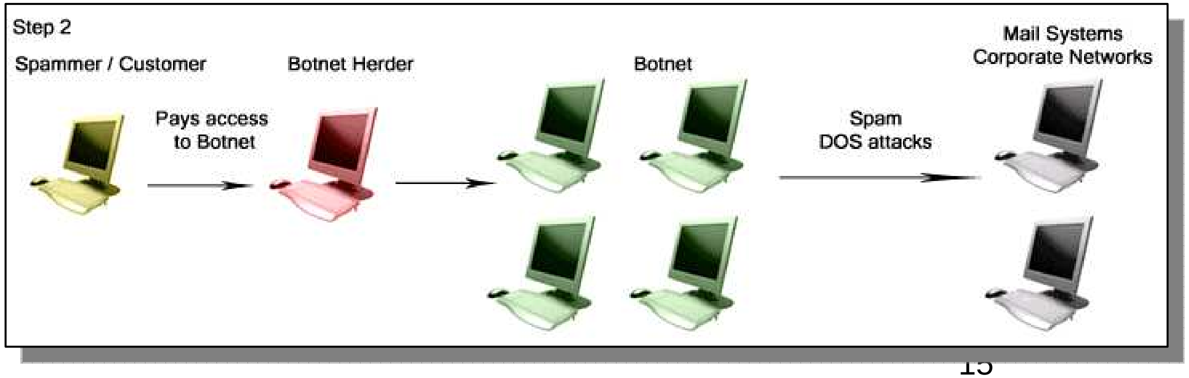
A network of compromised computers controlled by the **botmaster.**

Range in size from hundreds to millions of hosts.

Purpose varies: denial of service attacks, spam delivery, stealing banking credentials, stealing data, etc.

Typically runs hidden from the user and utilizes a command and control structure, through IRC, HTML, SSL, Twitter, IM or custom-built solutions.

Hosts can be infected by drive by downloads from malicious or compromised websites, executables delivered through email or web, as well as malicious PDF and Word files.



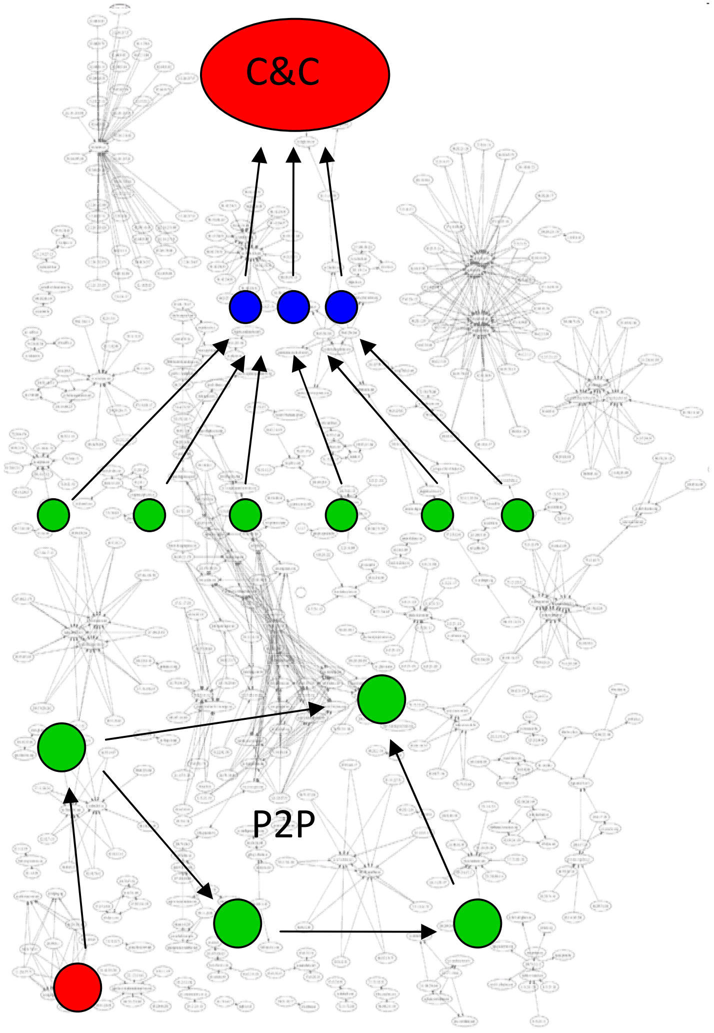
Botnet Communication Architectures

Collection of infected hosts used to send spam, etc.

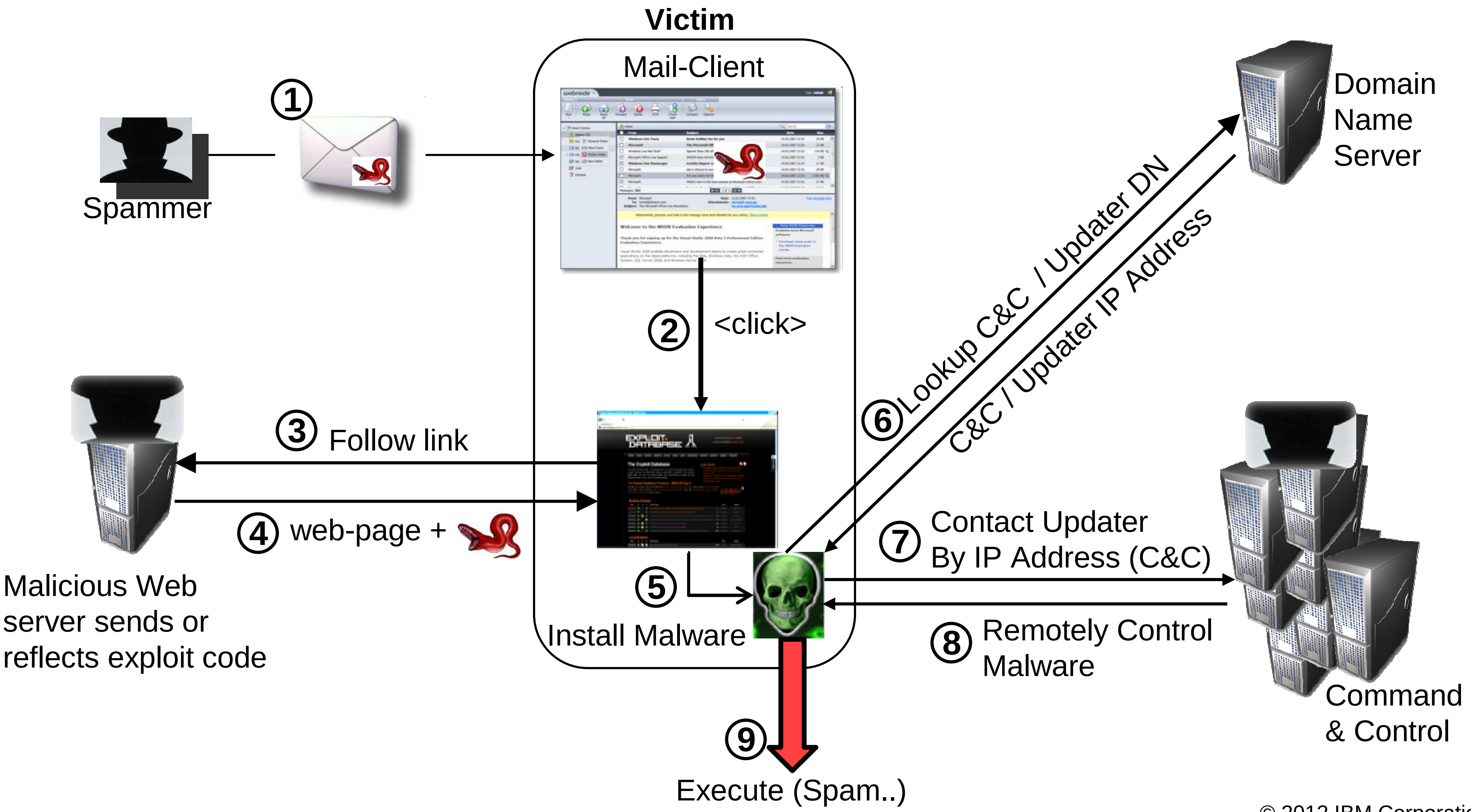
Bots connect to C&C (command & control) hosts

Botnets are becoming more sophisticated and harder to track – peer-to-peer, fast fluxing, (distributed) vs. hierarchical control structure

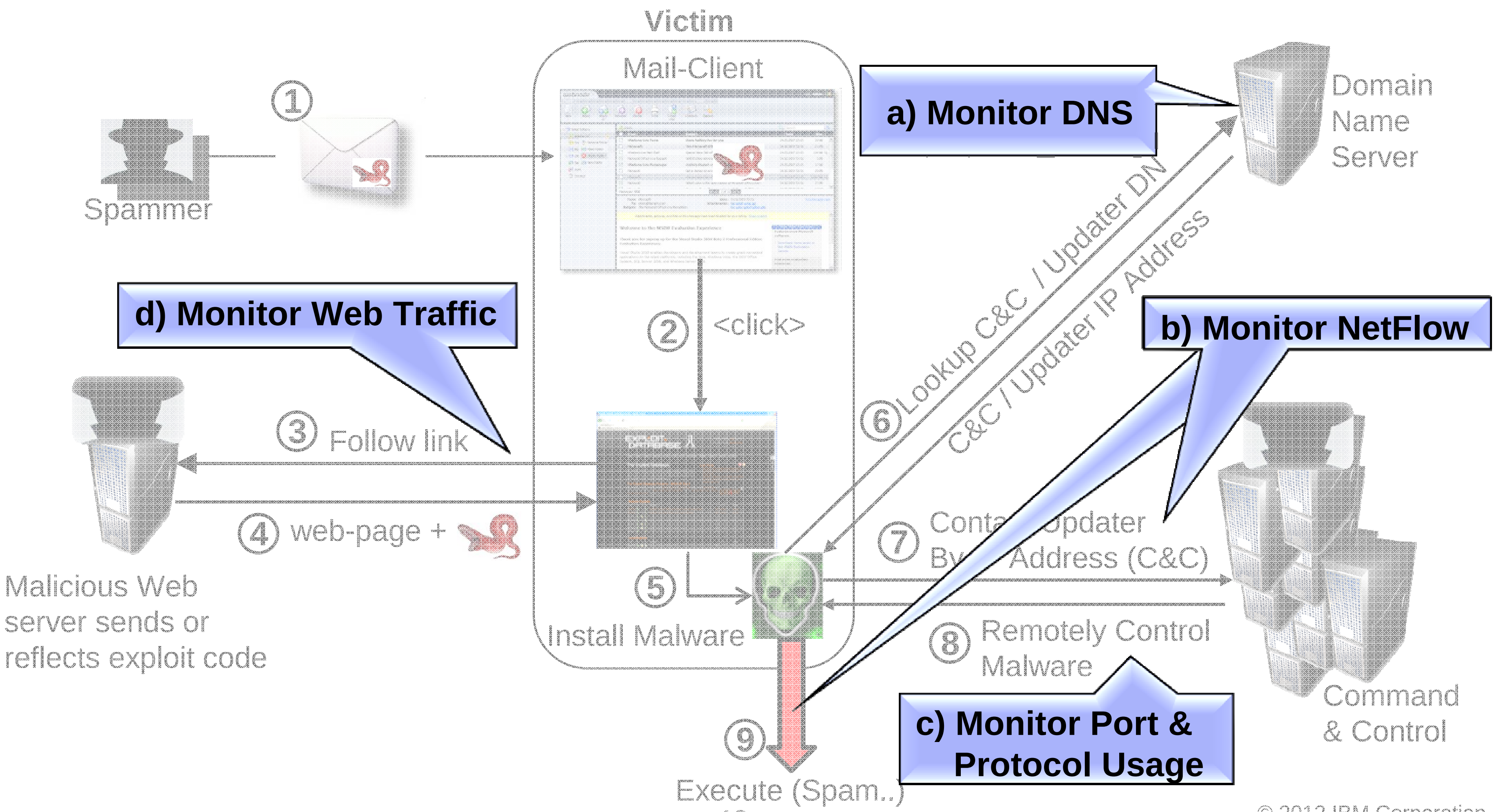
Hidden communications



Threat Example



Threat Example



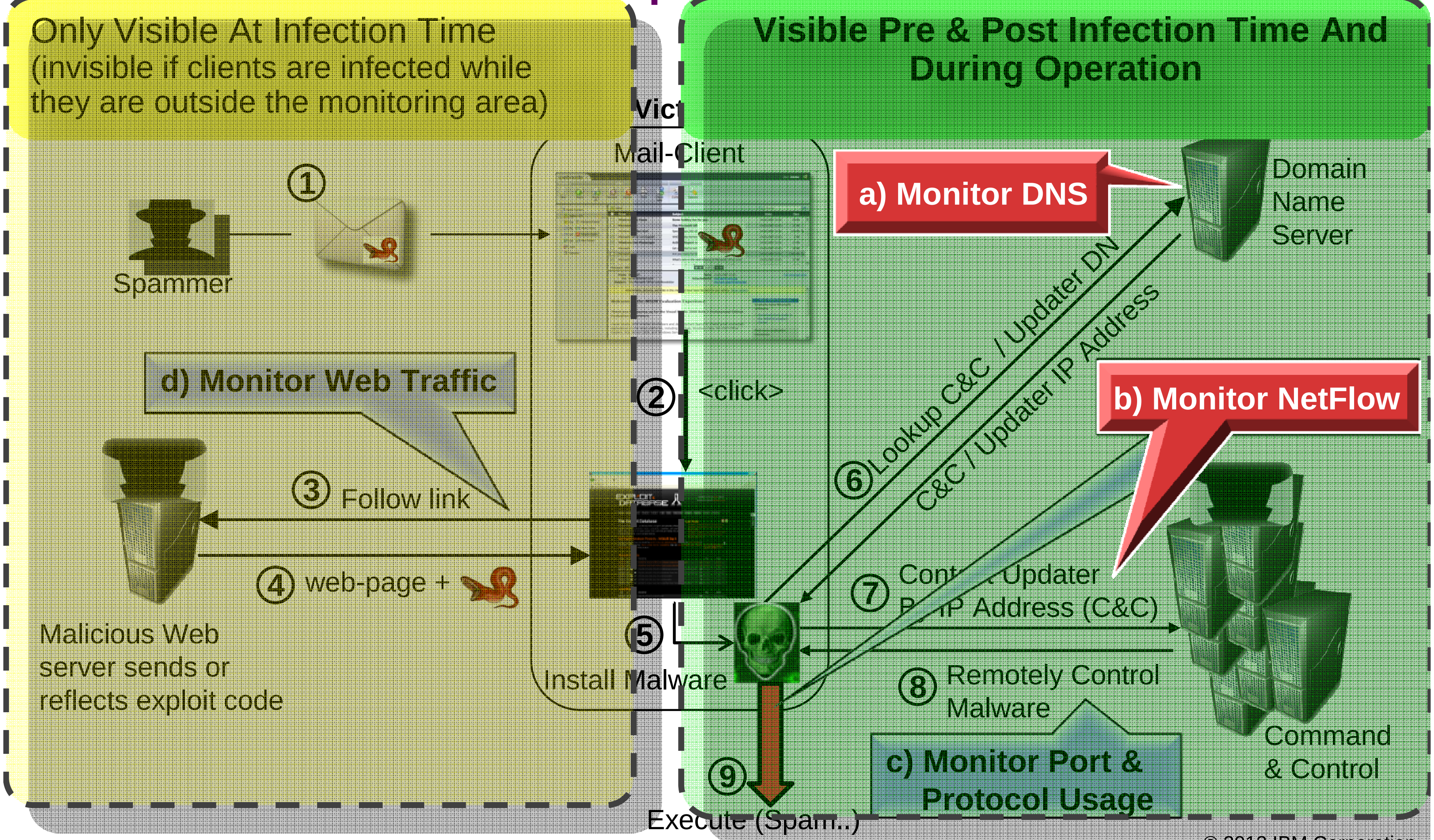
a) Monitor DNS

b) Monitor NetFlow

c) Monitor Port & Protocol Usage

d) Monitor Web Traffic

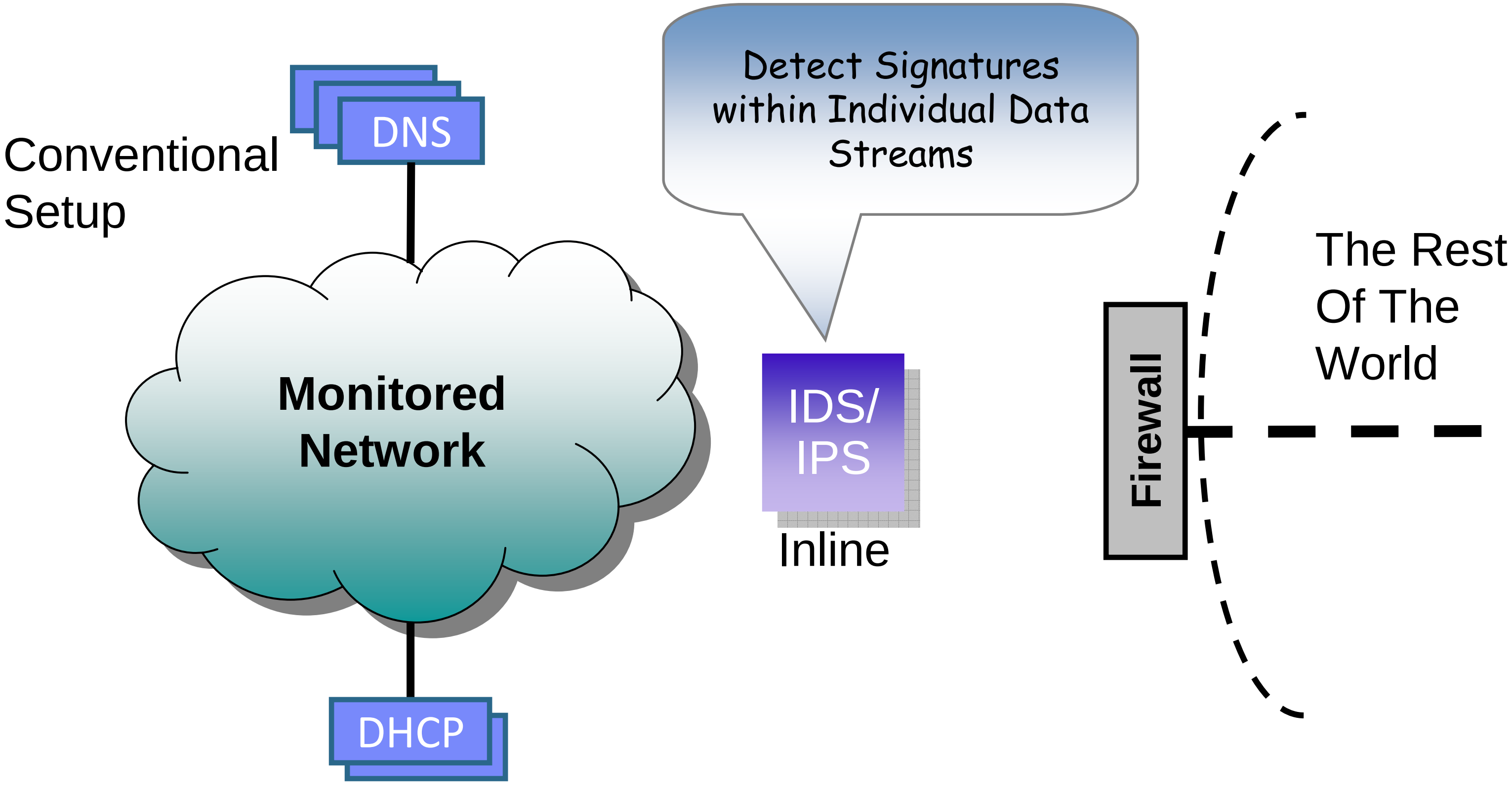
Threat Example



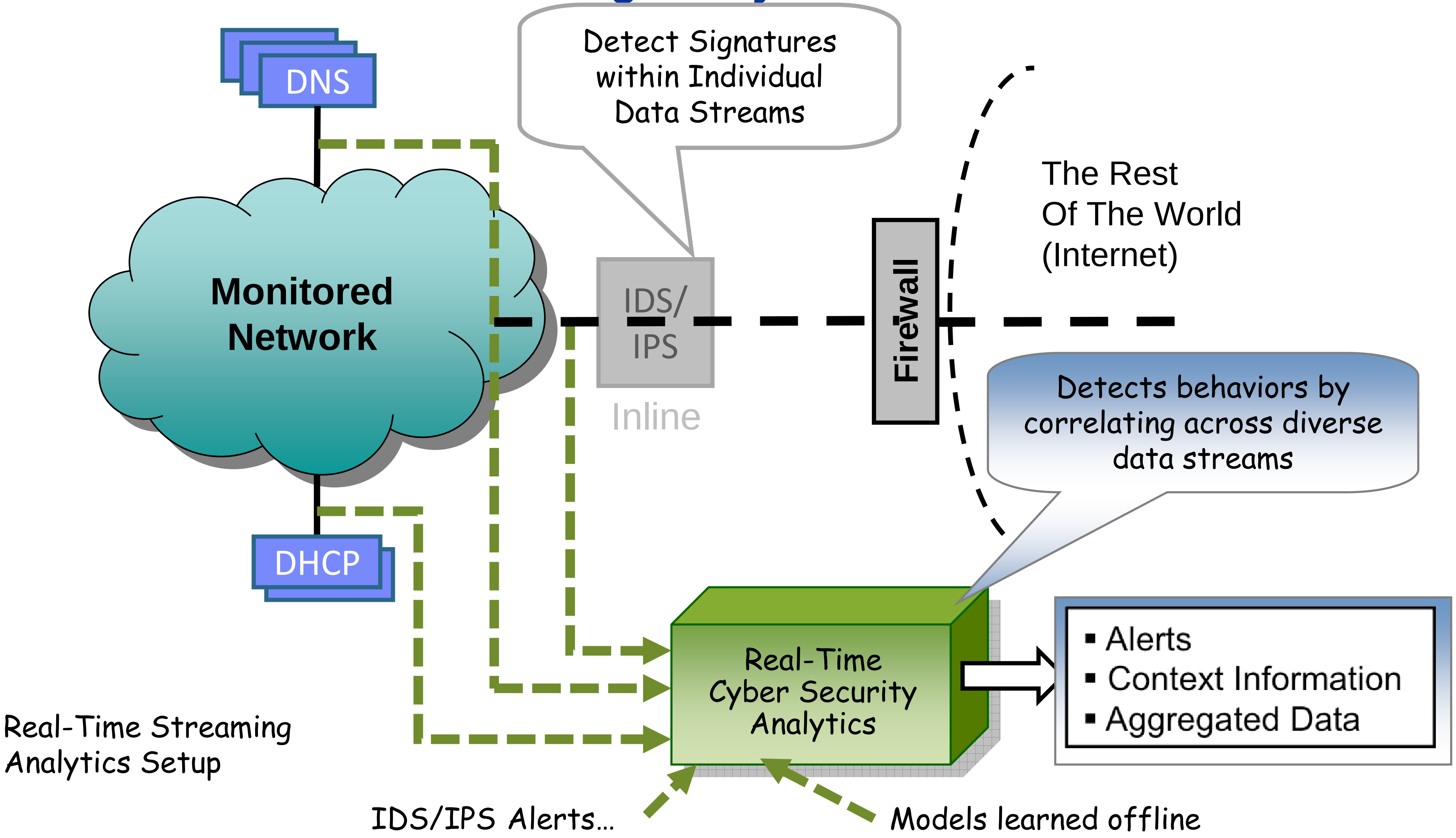
Only Visible At Infection Time
(invisible if clients are infected while they are outside the monitoring area)

Visible Pre & Post Infection Time And During Operation

Traditional Security Analytics

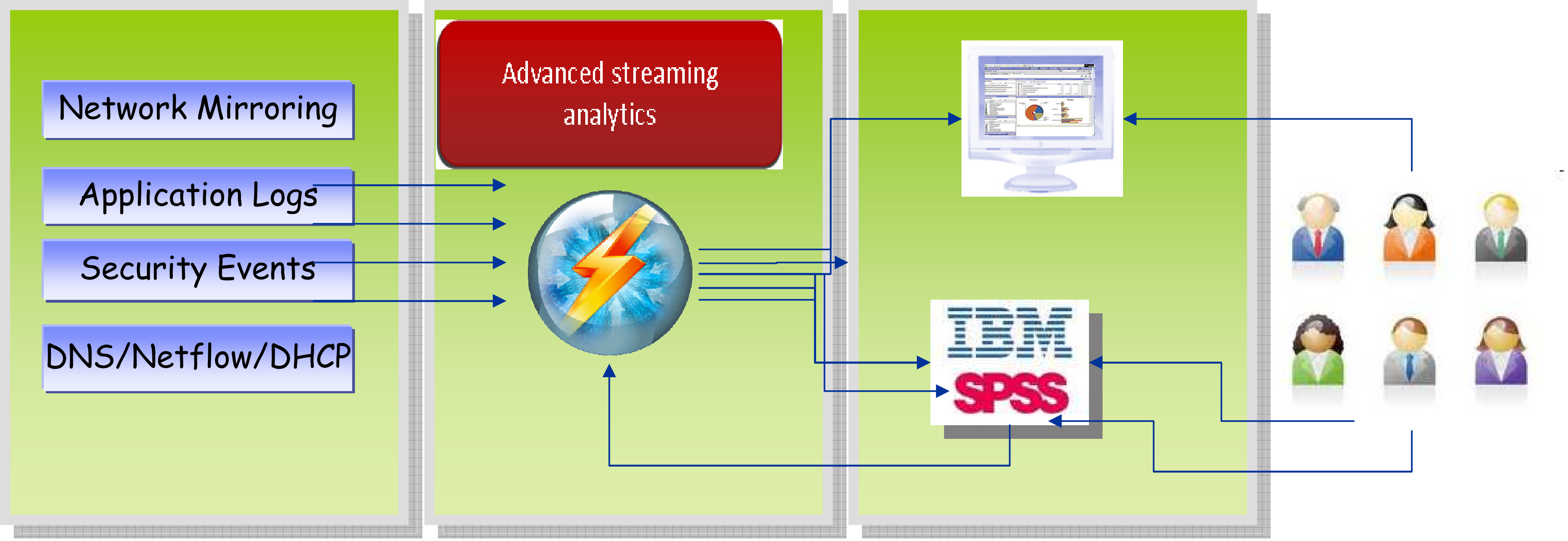


Streaming Analytics


Detect Signatures within Individual Data Streams
DNS
Monitored Network
DHCP
IDS/ IPS
Inline
Firewall
The Rest Of The World (Internet)
Detects behaviors by correlating across diverse data streams
Real-Time Cyber Security Analytics
Real-Time Streaming Analytics Setup
IDS/IPS Alerts…
Models learned offline

- Alerts
- Context Information
- Aggregated Data

Solution Analytics Lifecycle Overview

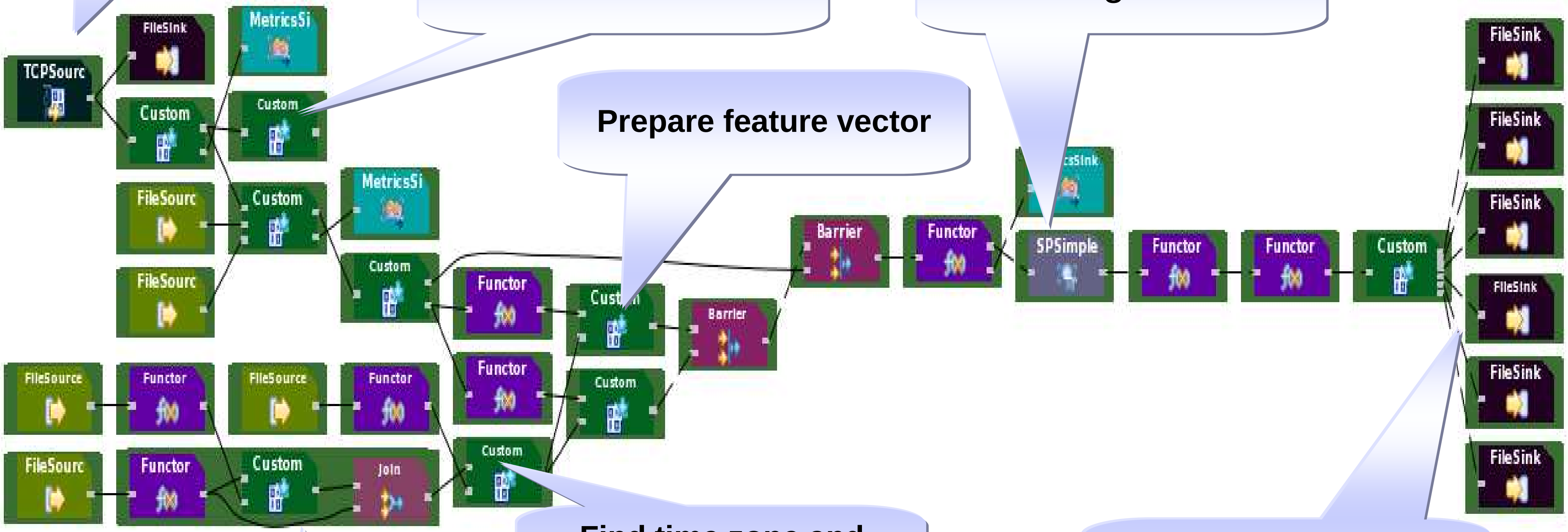


Get DNS traffic

Ingest DNS responses

Run algorithms

Prepare feature vector

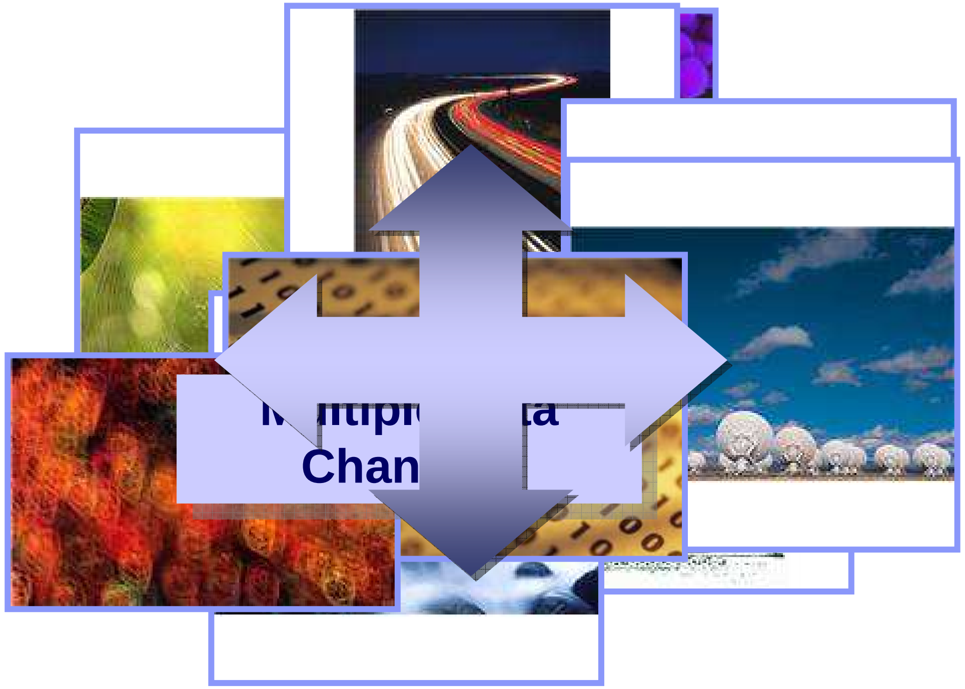


Find time zone and calculate entropies

Send out insights

Load geo-spatial data

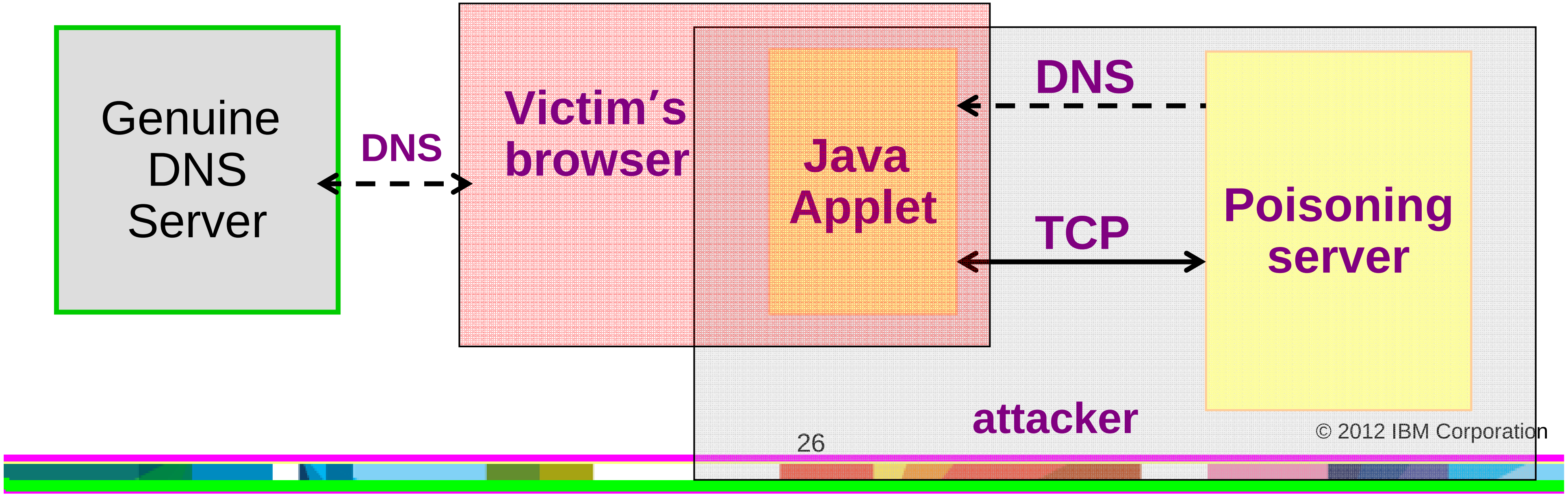
Streams Real Time Analytics
For Cyber Security



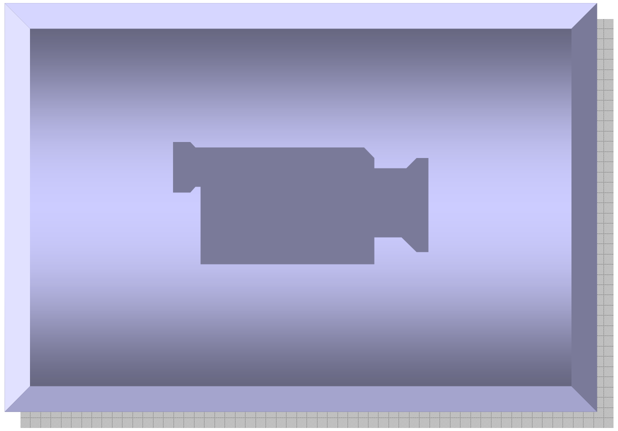
DNS Cache Poisoning Demo

DNS Cache Poisoning Demo Scenario

- ⌘ Attack is initiated upon visiting a malicious web page
- ⌘ Using the applet, bind on all UDP ports, leave one port open (65534)
- ⌘ Load a set of hosts to poison
- ⌘ Loop until success:
  - Get next host
  - Notify to poisoning server to target the loaded host
  - Generate a DNS query using the browser
  - Validate success
- ⌘ Load actual payload



DNS Cache Poisoning Demo



Responding to Cyber Threats

Cyber Security Challenges

High Volume Data Streams

Threats emerging at high rate

- Short lived patterns
- New threats

Evasive threats hard to detect

- Discriminative only across multiple channels (DNS + Raw Traffic + …)
- Slow and low threats

Cost

- Adaptation to changing data rates, extensibility
- Domain experts hard to find and very expensive
- Leveraging existing capabilities

Streams Analytics Solution

Scalable Real-Time Analytics Platform

Real-time detection

- Behavioral models covering many variants
- Flexible analytics
- Analytics across multiple types of data, including DNS, Raw Traffic, Alerts, Access
- Offline/On-line Models based analytics

Programmable and extensible platform

- Real-time detection enabling quick response
- Scalable, self-managing analytics middle-ware
- Domain knowledge easily translates into analytics applied broadly across all traffic

Real-time Intelligence Generation

Real-time Intelligence Generation

- Determine who is communicating to whom and how they are communicating
- Find what people think about a certain person, organization or company
- Find interests, activities, locations, etc. about individuals or groups
- Specifically monitor activities of persons in a blacklist
- Find other suspicious content

Real-time Intelligence Generation

From:

- Different kinds of social media like Facebook, Youtube, IRC, Twitter, etc.
- Interception of traffic to and from web-based applications
- Crawling or publicly available APIs

Using:

- Scalable network and “data-in-motion” analytics platforms
- Advanced analytics technologies (unstructured data analytics, real-time, predictive analytics)

Content monitoring in text in different protocols

- Look for keywords (in English, Arabic,…) in text on HTTP, FTP, SMTP, IRC, etc.

Suspect monitoring in Internet traffic

- Look for actions by suspects (identified by Facebook ID, Yahoo email address, etc.)

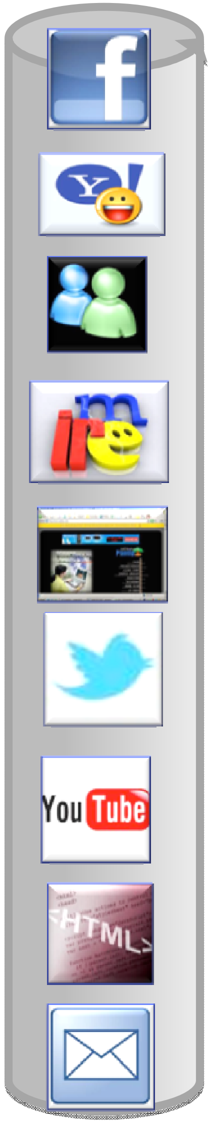
Real-time streaming social network synthesis and analysis

- Generate social networks from intercepted Facebook communications
- Explore the social network and interactions between people

Real-time sentiment analysis

- Domain-specific or domain-independent (English)

Challenge in Two Dimensions – Lots of data and lots of sources



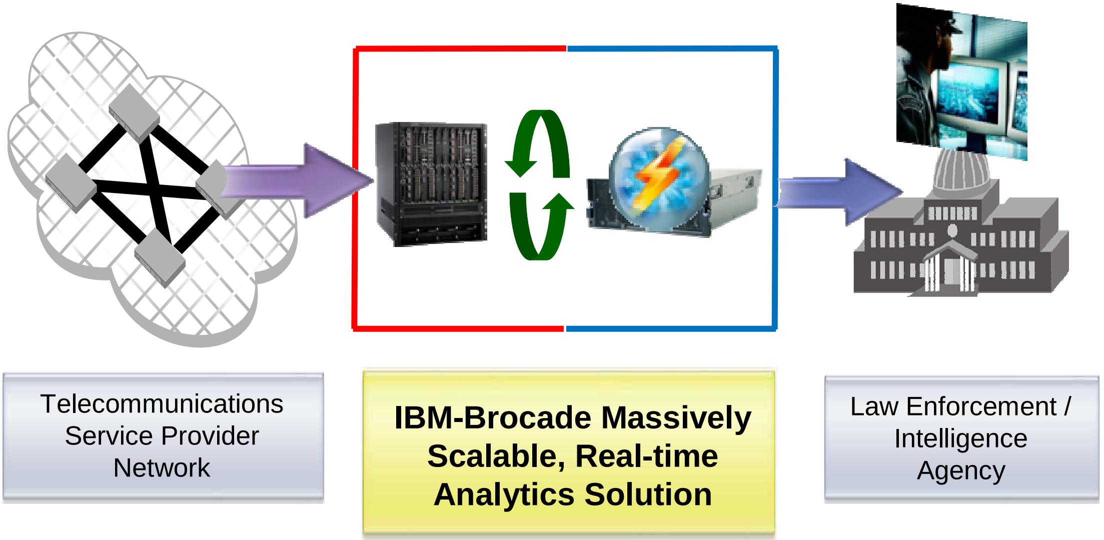
What you need:

- A massively parallel platform
- Easily extensible
- Self-healing
- Filtering raw data for relevant intelligence
- Correlating data from different sources
- Allow deployment of analytical/predictive models

- **Variety of data sources:**
  - Network Protocols like HTTP, SMTP, FTP, SIP, IRC, DNS, etc.
  - Web-based applications like social networking apps, email, chat, etc.
  - Sensor networks for surveillance and environment monitoring
  - Need to monitor different kinds of threats

Delivering real-time intelligence

- Anomaly Detection - Identify unknown security threats; advanced persistent threats; cyber attacks; worms, botnets; behavioral based modeling
- Adaptive intelligence gathering - Manage dynamic blacklists; monitor for suspicious patterns in content and actions by known or unknown suspects
- Situational awareness – Map who is talking to whom; create behavioral profiles
- Assist law enforcement – Correlate multiple data streams; aggregate time sensitive information; perform monitoring and deep packet analysis

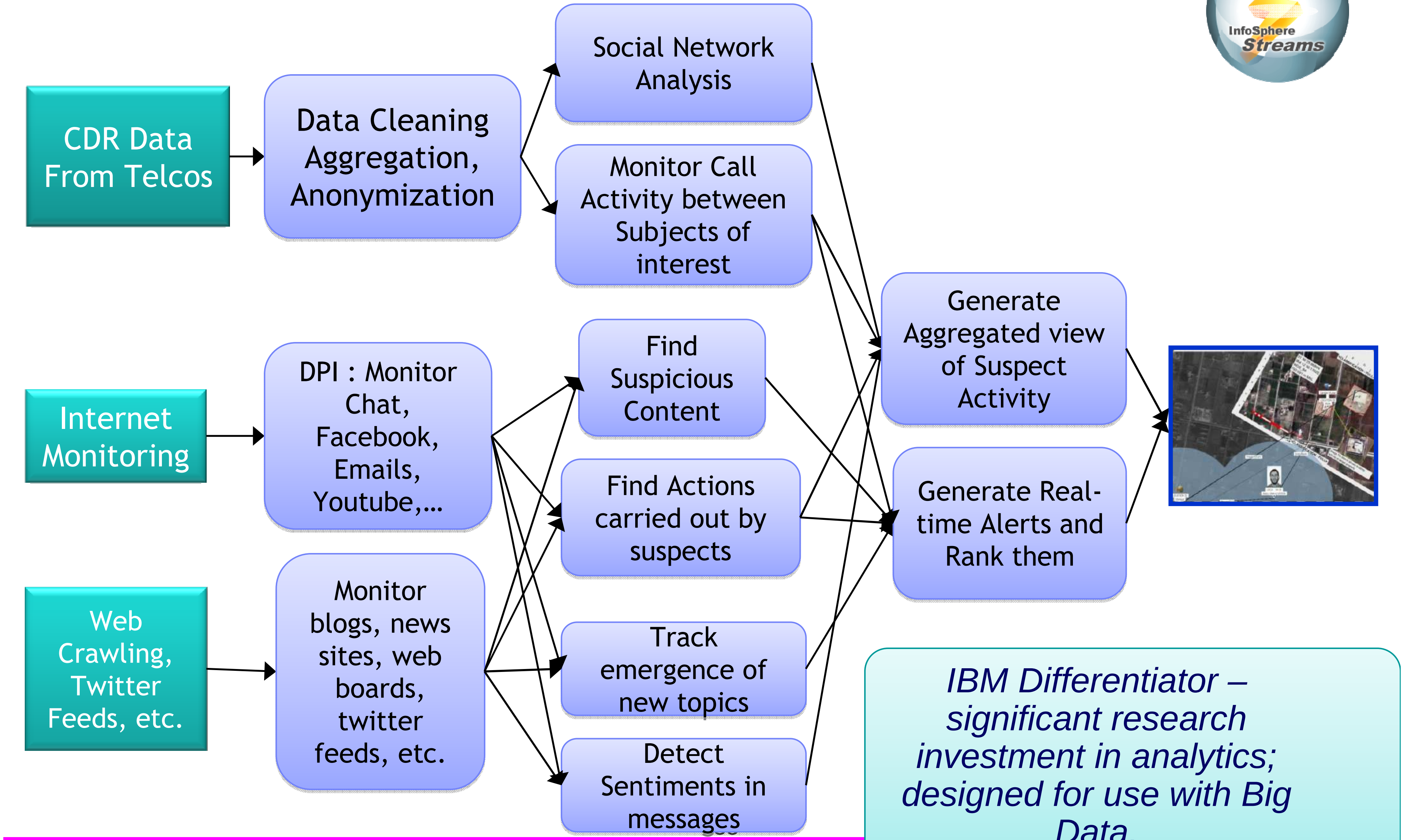


Telecommunications Service Provider Network

IBM-Brocade Massively Scalable, Real-time Analytics Solution

Law Enforcement / Intelligence Agency

Streams Processing Flows Sample



CDR Data From Telcos

Data Cleaning Aggregation, Anonymization

Social Network Analysis

Monitor Call Activity between Subjects of interest

Internet Monitoring

DPI : Monitor Chat, Facebook, Emails, Youtube,…

Find Suspicious Content

Find Actions carried out by suspects

Web Crawling, Twitter Feeds, etc.

Monitor blogs, news sites, web boards, twitter feeds, etc.

Track emergence of new topics

Detect Sentiments in messages

Generate Aggregated view of Suspect Activity

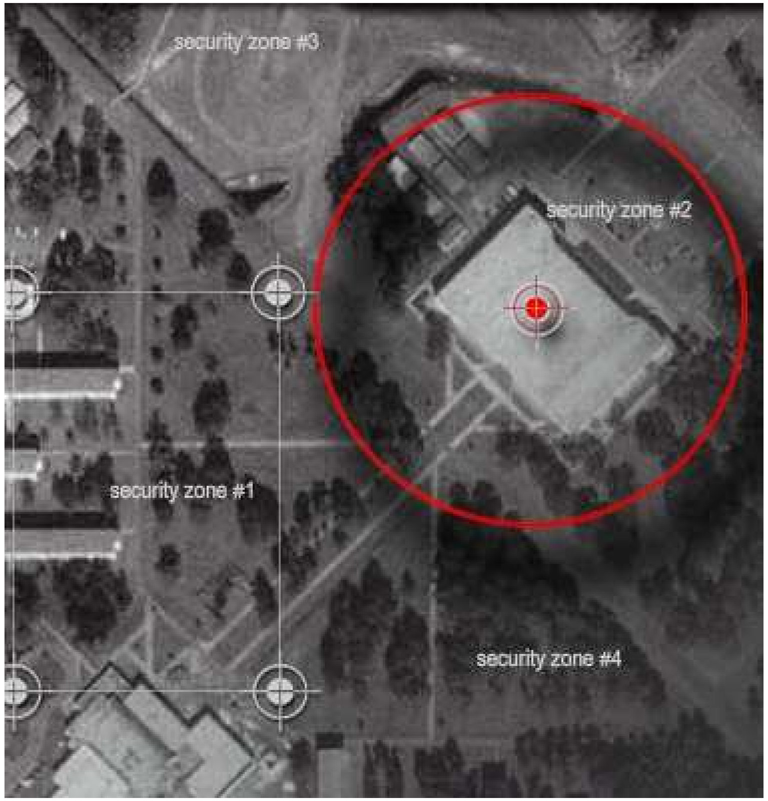
Generate Real-time Alerts and Rank them

IBM Differentiator – significant research investment in analytics; designed for use with Big Data.

TerraEchos Adelos™– Covert Intrusion Detection

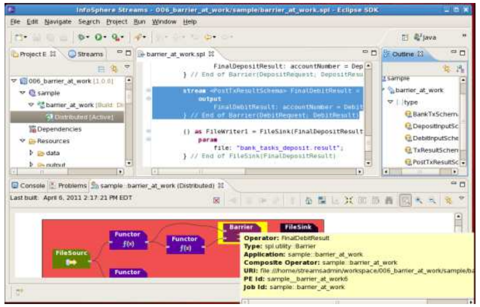
State-of-the-art covert surveillance based on Streams platform

Acoustic signals from buried fiber optic cables are monitored, analyzed and reported in real time to locate intruders



The InfoSphere Streams Platform

Streams Processing Language and IDE



InfoSphere Streams - 006_barrier_at_work/sample/barrier_at_work.spl - Eclipse SDK

```
        FinalDepositResult: accountNumber = Dep
    } // End of Barrier(DepositRequest; DepositResu

    stream <PostTxResultSchema> FinalDebitResult =
      output
        FinalDebitResult: accountNumber = Debit
    } // End of Barrier(DebitRequest; DebitResult)

    () as FileWriter1 = FileSink(FinalDepositResult
      param
        file: "bank_tasks_deposit.result";
    } // End of FileSink(FinalDepositResult)
```

Last built: April 6, 2011 2:17:21 PM EDT

Operator: FinalDebitResult
Type: spl.utility::Barrier
Application: sample::barrier_at_work
Composite Operator: sample::barrier_at_work
URI: file:///home/streamsadmin/workspace/006_barrier_at_work/sample/ba
PE Id: sample__barrier_at_work6
Job Id: sample::barrier_at_work

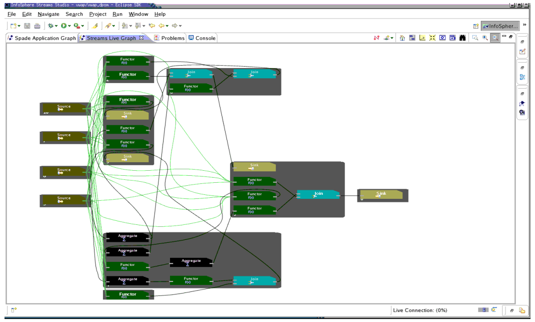
eclipse

Streams Studio
Eclipse IDE for SPL

Runtime Environment

Scalable stream processing runtime

Tools and Technology Integration



DB2 IDS solidDB WebSphere Front Office 3.0 ORACLE FIX PROTOCOL INDUSTRY-DRIVEN MESSAGING STANDARD Microsoft SQL Server MySQL

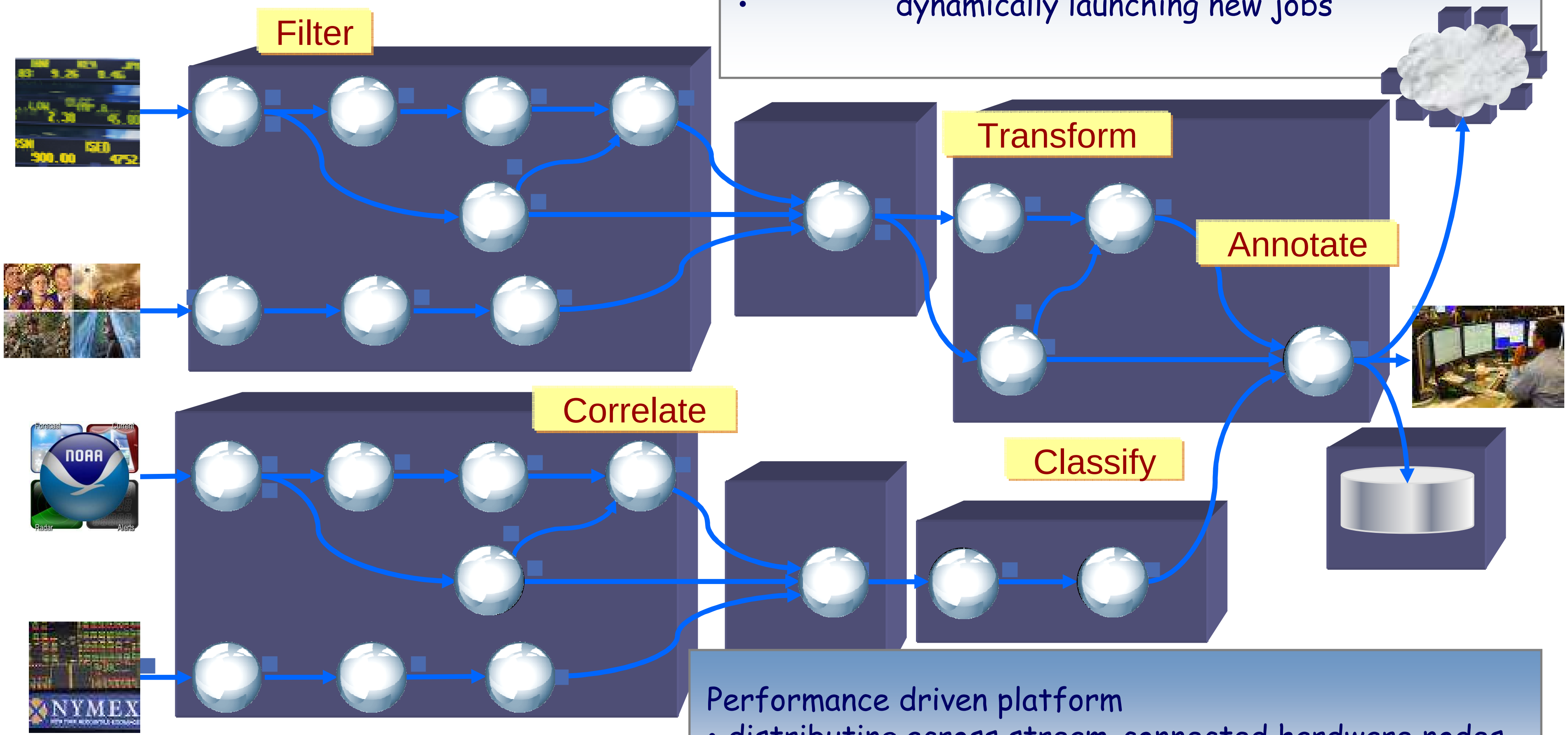
Streamsight,
Built-in Stream Relational Analytics,
Adapters, Toolkits

Supported on x86 hardware, RedHat Enterprise Linux Version 5 (5.3 and up)

- continuous ingestion
- continuous analysis

infrastructure provides services for

- developing stream processing applications
- scheduling analytics across h/w nodes
- dynamically launching new jobs



Performance driven platform

- distributing across stream-connected hardware nodes
- “fuses” elements together for lower communication latency

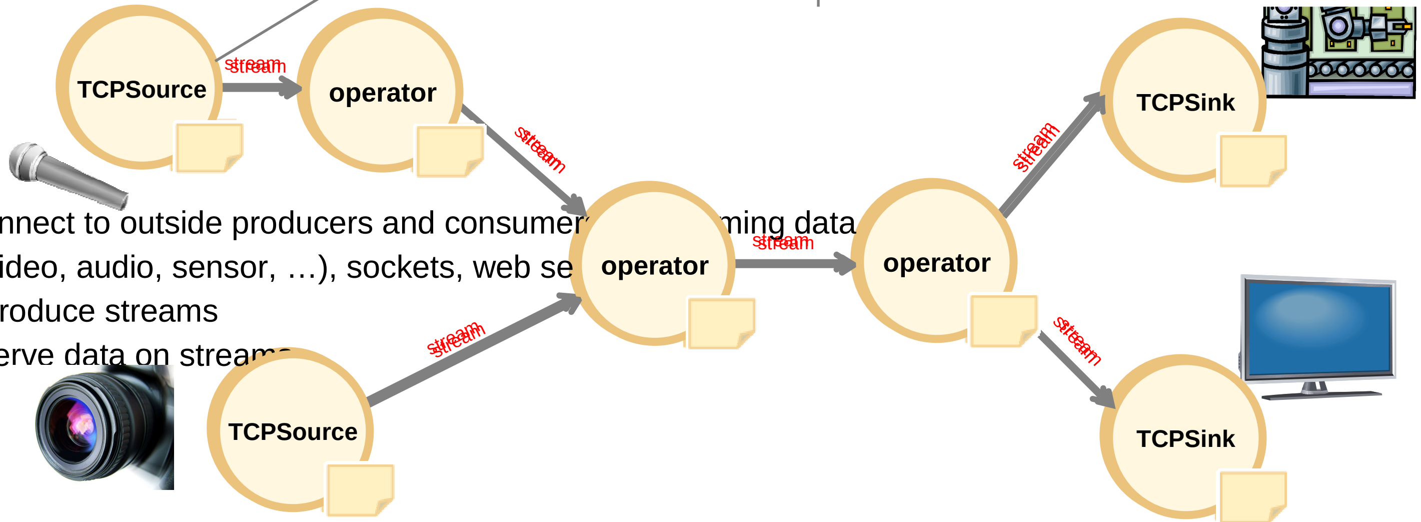
A stream processing program/job is a data-flow network

Streams are written/produced by ***operators***

Operators produce streams by

- observing data (tuples) on their input streams
- performing some kind(s) of computation
- writing data (tuples) to their output streams

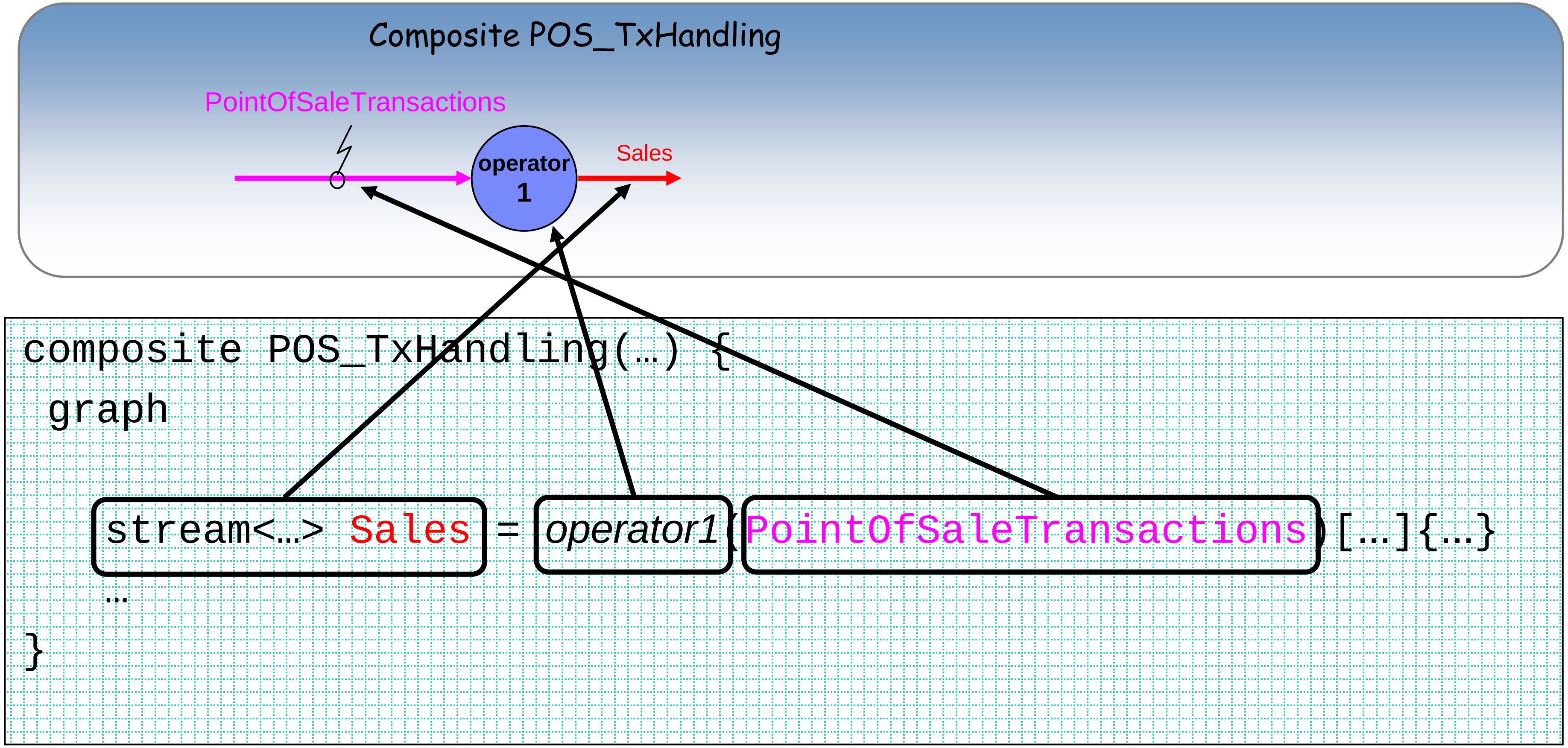
Not all Streams jobs contain edge adaptors. Some streams can be conveyed between Streams jobs by using export and import operators



Edge adapters connect to outside producers and consumer ming data

- devices (video, audio, sensor, …), sockets, web se
- Sources produce streams
- Sinks observe data on stream

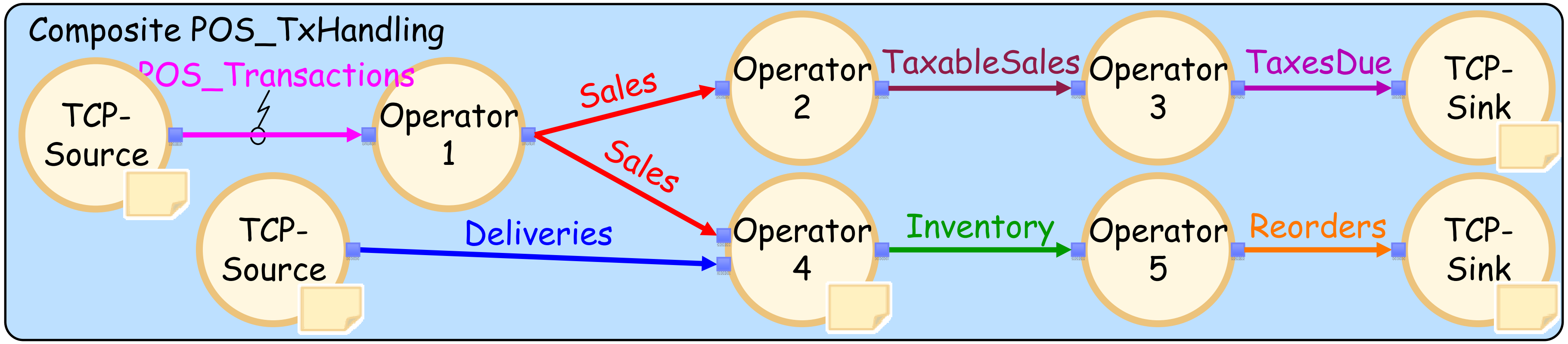
Expressing a flow composition with stream definitions



```
composite POS_TxHandling(…) {
 graph

    stream<…> Sales = operator1(PointOfSaleTransactions)[…]{…}
    …
}
```

Composing a Flow Graph with Stream Definitions



```
composite POS_TxHandling
{
    graph
        stream<…> POS_Transactions = TCPSource() {…}
        stream<…> Sales = Operator1(POS_Transactions) {…}
        stream<…> TaxableSales = Operator2(Sales) {…}
        stream<…> TaxesDue = Operator3(TaxableSales) {…}
        () as Sink1 = TCPSink(TaxesDue) {…}
        stream<…> Deliveries = TCPSource() {…}
        stream<…> Inventory = Operator4(Sales;Deliveries) {…}
        stream<…> Reorders = Operator5(Inventory) {…}
        () as Sink2 = TCPSink(Reorders) {…}
}
```

SPL Standard Toolkit operators (Included with Streams)

Relational Operators

Filter
Functor
Punctor
Sort
Join
Aggregate

Utility Operators

Custom
Beacon
Throttle
Delay
Barrier
Pair

Split
DeDuplicate
Union
ThreadedSplit
DynamicFilter
Gate

Adapter Operators

FileSource
FileSink
DirectoryScan
TCPSource
TCPSink

UDPSource
UDPSink
Export
Import

SPL Standard Toolkit operators – Relational Operators

Filter

create an output stream with subset of input tuples

Functor

Add attributes, remove attributes, filter tuples, map output attributes to input attributes

Punctor

Punctuation mark to delimit windows

Sort

Window-based sorting

Join

The Join operator is used to correlate tuples from two streams based on user-specified match and window configurations.

Aggregate

Window-based aggregates, with group by

SPL Standard Toolkit operators – Adaptor Operators

FileSource

reads data from a file and produces tuples

FileSink

writes tuples to a file

DirecotoryScan

watches a directory, generates file names, one per new file found

TCPSource, UDPSource

reads data from a TCP/UDP socket and creates tuples

TCPSink, UDPSink

writes tuples to a TCP/UDP socket

Export

sends a stream from the current application, making it available to Import operators of applications running in the same streaming middleware instance

Import

receives tuples from streams made available by Export operators of applications running in the same streaming middleware instance

MetricsSink

creates Custom Operator Metrics and updates them with values when a tuple is received

SPL Standard Toolkit operators – Utility Operators

Custom

special operator that can submit tuples from within its onTuple and onPunct clauses.

Beacon

a utility source that generates tuples on-the-fly

Throttle

paces a stream to make it flow at a specified rate

Delay

delays a stream by a given amount while keeping the inter-arrival times of tuples and punctuations intact

Barrier

synchronize tuples from two or more streams /synchronizing the results from performing parallel tasks on the same stream (to one)

Pair

pairs tuples from 2 or more streams (same schema)

Split

splits a stream into one or more output streams

DeDuplicate

suppresses duplicate tuples seen within a given time period

Union

combines tuples from streams connected to different input ports

ThreadedSplit

splits tuples across multiple output ports to improve concurrency

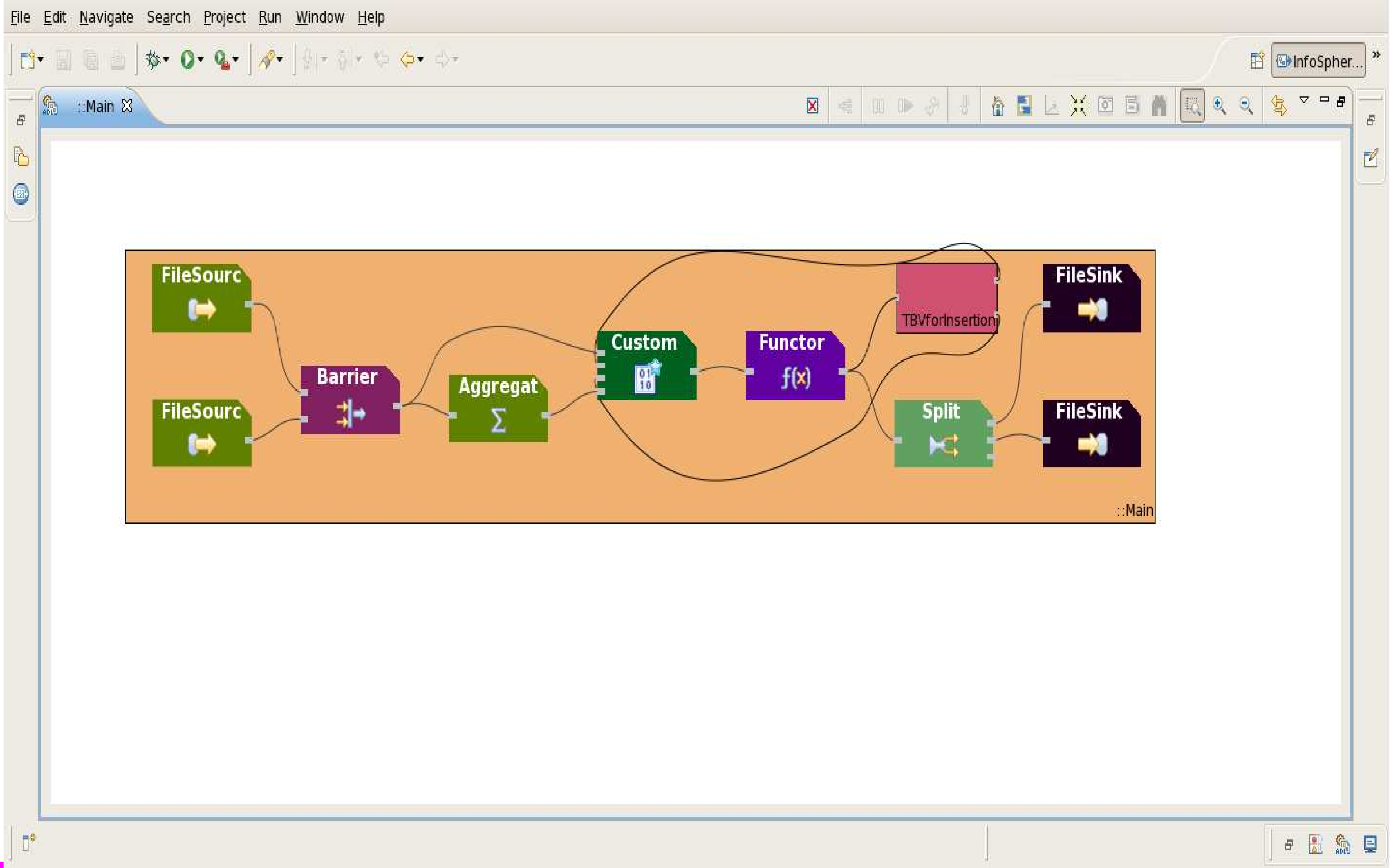
DynamcFilter

Filter based on runtime criteria

Gate

controls the rate at which tuples are passed through

Streams Studio Integrated Development Environment



Streams Objects: Runtime View

Instance

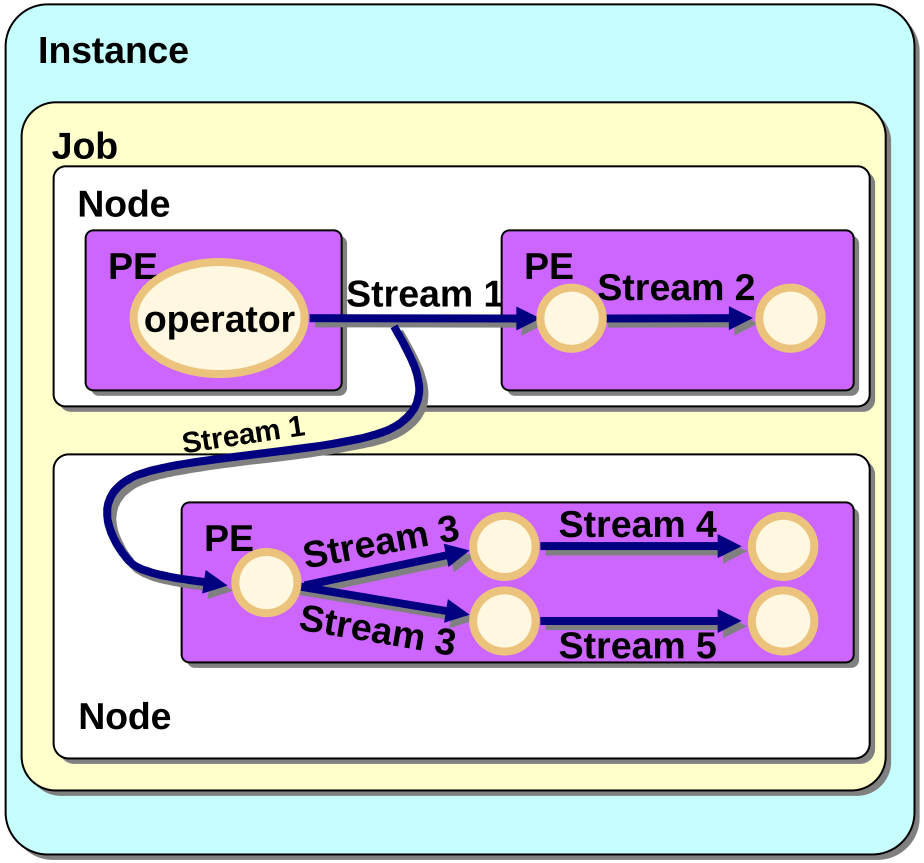
- Runtime instantiation of InfoSphere Streams executing across one or more hosts
- Collection of components and services

Processing Element (PE)

- Fundamental execution unit that is run by the Streams instance
- Can encapsulate a single operator or many “fused” operators

Job

- A deployed Streams application executing in an instance
- Consists of one or more PEs


Instance
Job
Node
PE
operator
Stream 1
PE
Stream 2
Stream 1
PE
Stream 3
Stream 4
Stream 3
Stream 5
Node