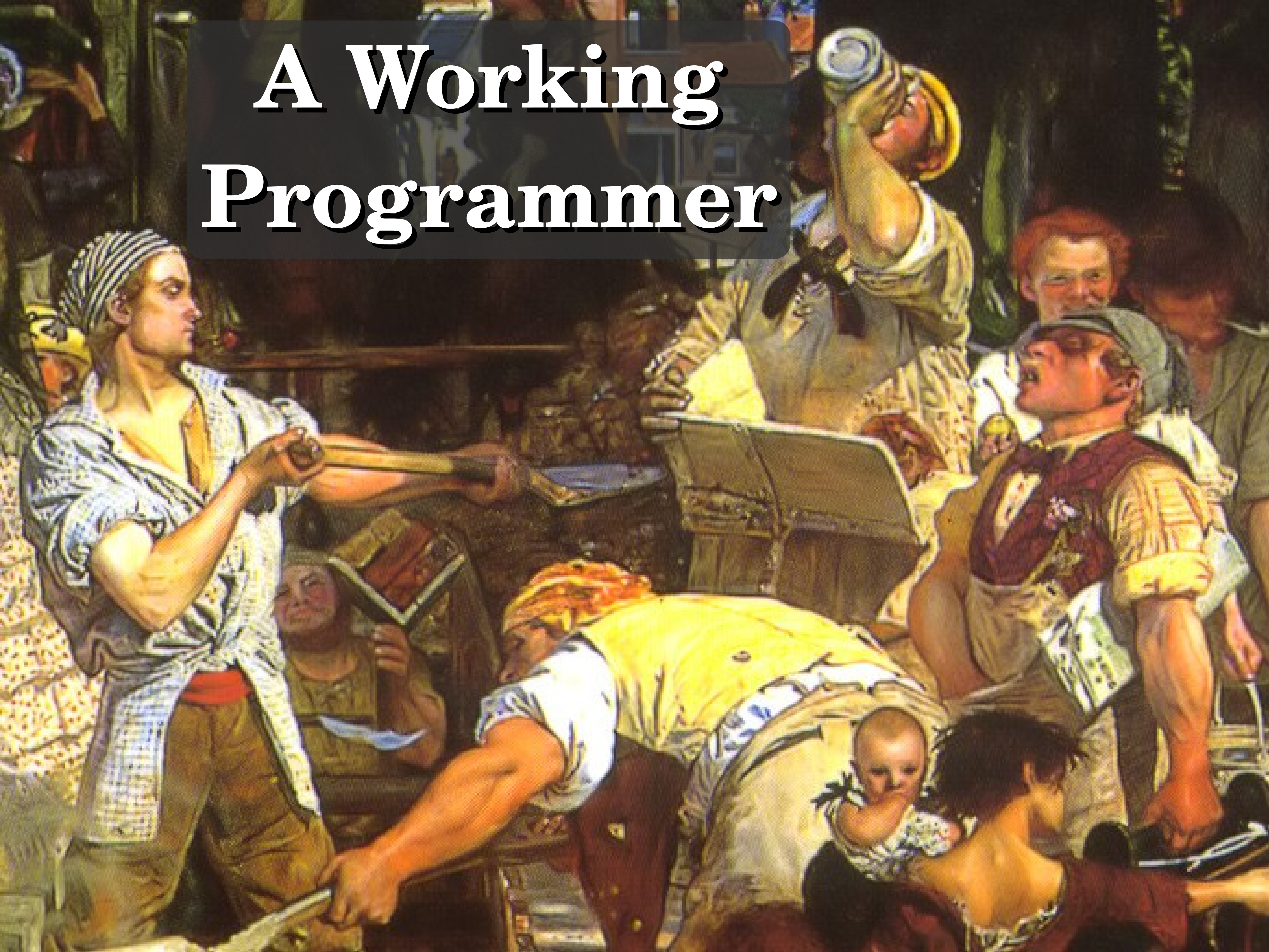
SQL and `core.logic` Killed My ORM

Craig Brozefsky
@cbrozefsky

ThreatGRID
Malware Analysis & Threat Intelligence

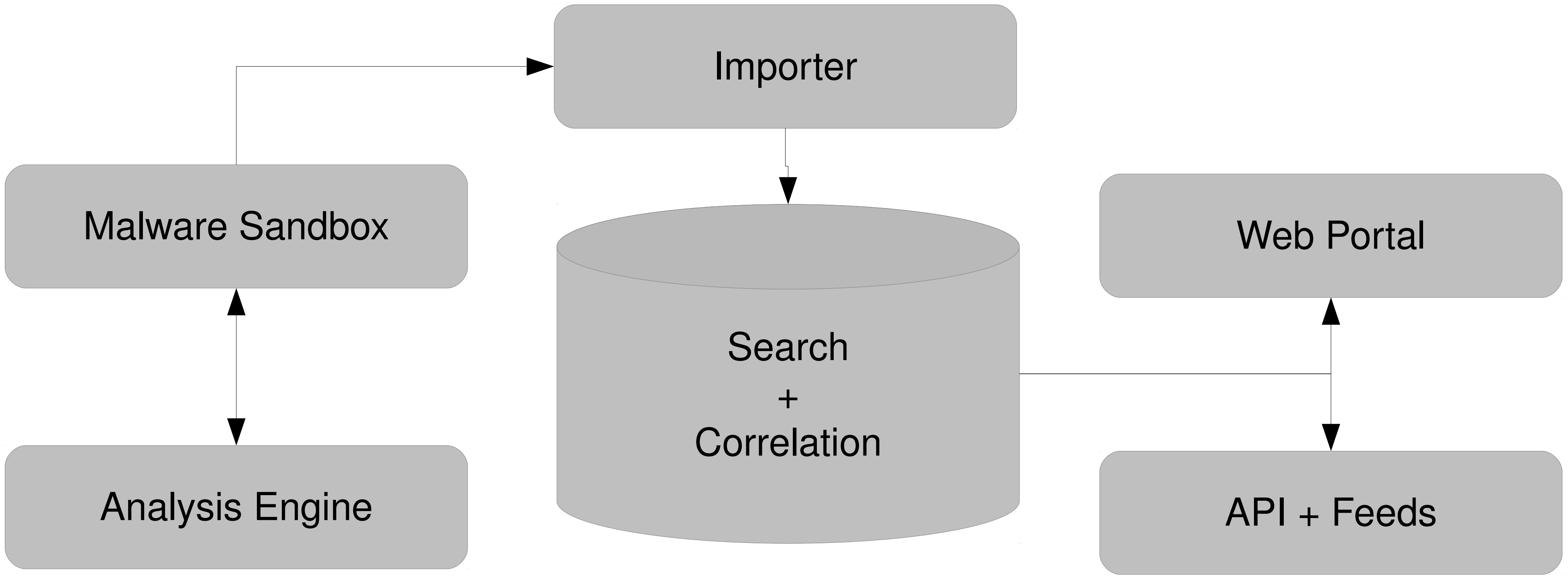
A Working Programmer



Not A Hater

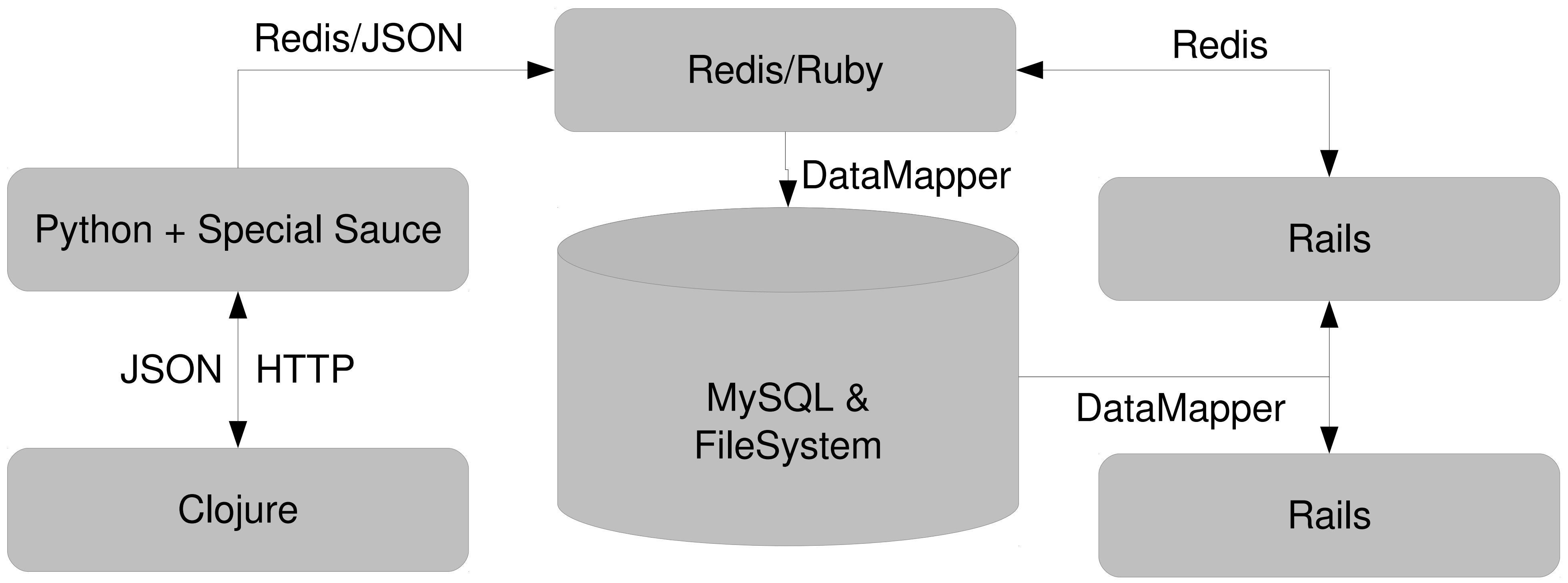
ThreatGRID

Malware Analysis & Threat Intelligence

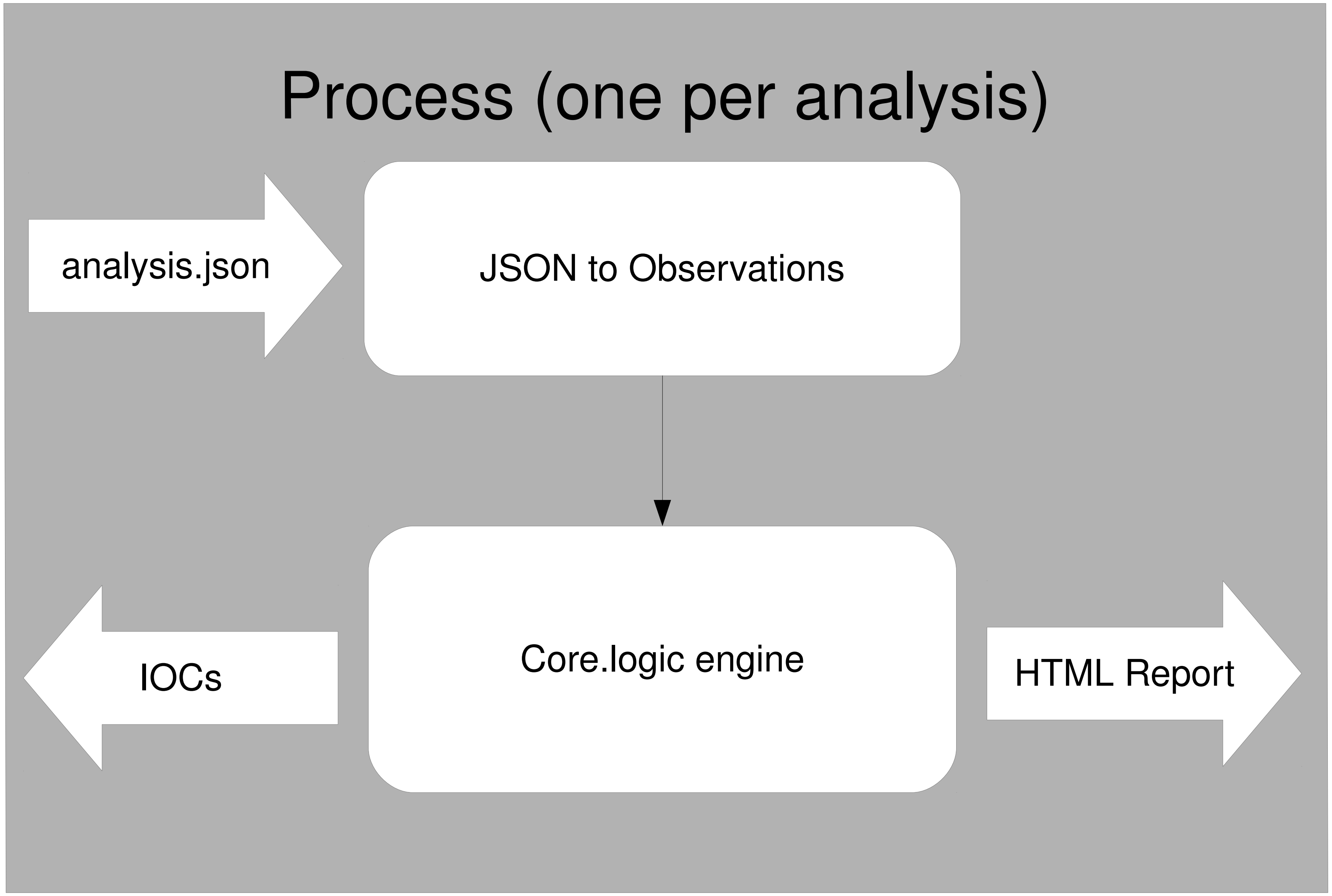


The Production Prototype

Technology Overview



Malware Analysis Engine



Observations

- wrapper around defrel
- field tags and indexing
- field naming convention
- metadata
- can be generated from queries
- ~100 in current model

An Observation Example

```
(defobs process-modified-path
  [#^{:tag Integer :required true :index true} pid
   #^{:tag Pathname :required true :index true
      :doc "A filename or a directory." } path]
  :doc "A pathname modified by a process associated by the PID."
       :scope [:sample]
  :tags ["process" "file" "directory" "path"])
```

IOCs

- Both core.logic programs and goals
- severity and confidence
- explanation and suggested remediation
- some metadata

An IOC Example

```
(defioc network-downloaded-executable
  :title "Downloaded PE Executable"
  :description "A PE executable was downloaded over the network... ."
  :severity 80 :confidence 95
  :category ["file", "network", "artifact"] :tags ["dropper"]
  :variables [IP Port Protocol Network_Stream Artifact_ID]
  :query
  ((artifact Artifact_ID "network" (lvar) (lvar) (lvar))
   (artifact-relation Artifact_ID "network" Network_Stream)
   (artifact-type Artifact_ID "exe")
   (network-stream Network_Stream (lvar) (lvar) IP Port)
   (network-stream-protocol Network_Stream Protocol)))
```

IOC :query

(:query ioc) =>

((artifact **Artifact_ID** "network" (lvar) (lvar) (lvar))

(artifact-relation **Artifact_ID** "network" **Network_Stream**)

(artifact-type **Artifact_ID** "exe")

(network-stream **Network_Stream** (lvar) (lvar) **IP Port**)

(network-stream-protocol **Network_Stream Protocol**)))

IOC :query wrapped

(run* [result]

(artifact **Artifact_ID** "network" (lvar) (lvar) (lvar))

(artifact-relation **Artifact_ID** "network" **Network_Stream**)

(artifact-type **Artifact_ID** "exe")

(network-stream **Network_Stream** (lvar) (lvar) **IP Port**)

(network-stream-protocol **Network_Stream Protocol**))

(== result [Artifact_ID Network....]))

Analysis Output Example

```
{"category": [ "persistence", "obfuscation" ],
 "confidence": 95, "severity": 95,
  "title": "Process Modified an Executable File",
  "data": [{"Process_Name": "exp2.tmp",
              "Path": "...KB01194541.exe",
              "Process_ID": 1216 },
             {"Process_Name": "3639fc660db0f51982da6c675f254626.exe",
              "Path": "...KB01194541.exe",
              "Process_ID": 1272 }],
  "tags": ["executable", "file", "process" ],
  "ioc": "modified-executable",
  "description": "Malware will modify executables on a system, to hide logs or other
                    evidence...."}
```

Just the data

```
[{"Process_Name": "exp2.tmp",
  "Path": "...KB01194541.exe",
  "Process_ID": 1216 },
 {"Process_Name""35f254626.exe",
  "Path": "...KB01194541.exe",
  "Process_ID": 1272 }],
```

The Crisis

- Data Model Woes
- Import Scaling
- Web Portal Perf

Data Model Woes

- Premature reification is the root of all evil
- Malware doesn't follow standards
- ORMs complect data and queries
- Competing Data Models

Import Scaling

- Conflicting Transactions
- Artificial keys mean read/write import
- Constraints are expensive

Web Portal Performance

- Report Generation from JSON->HTML
- Search queries
- Thread blockage
- Ruby performance

Still, not a hater!

A Couch Driven Development

- Functional **relational** programming, duh!
- Core.logic defrels map to tables
- Korma allows for functional composition of SQL
- Unified Data Model across all of TG.

Observation as Table

```
CREATE TABLE  process_modified_path (
    sample BYTEA NOT NULL,
    pid NUMERIC  NOT NULL,
    path VARCHAR  NOT NULL
);
CREATE INDEX  process_modified_path_sample_idx  ON
process_modified_path  ( sample );
CREATE INDEX  process_modified_path_pid_idx  ON
process_modified_path  ( pid );
CREATE INDEX  process_modified_path_path_idx  ON
process_modified_path  ( path );
```

•

Functional SQL

- Decompose/Compose queries
- Tight control of sql generation
- Full access to PSQL data types

SQL Korma

```
(select* :submissions) =>

{:group [],
 :from [{:table"submissions"}],
 :joins [],
 :where [],
 :ent {:table "submissions"},
 :type :select,
 :alias nil,
 :options nil,
 :fields [:korma.core/*],
 :results :results,
 :table "submissions",
 :order [],
 :modifiers [],
 :db nil,
 :aliases #{}
}
```

- Queries are maps
- Loves →
- Build some queries for great win

Partial Queries

```
(defn sample-info-query []

  (-> (select* :submissions)

      (restrict-to-visible-sample *current-user*)

      (ensure-joined :sample_data
                     :submissions.sample
                     :sample_data.sample)

      (fields :submissions.sample
              :submissions.login

...
```

Composing joins

```
(defn ensure-joined [query table lk fk]
  (if (joined? query table)
    query
    (kql/join query table (= lk fk))))
```

Composing Conditions

```
(defn restrict-to-path [query path
                        & {:keys [key]
                           :or {key :path}}]
  (if-not path
    query
    (-> query
        (kql/where
          (like key (escape-search-term path))))))
```

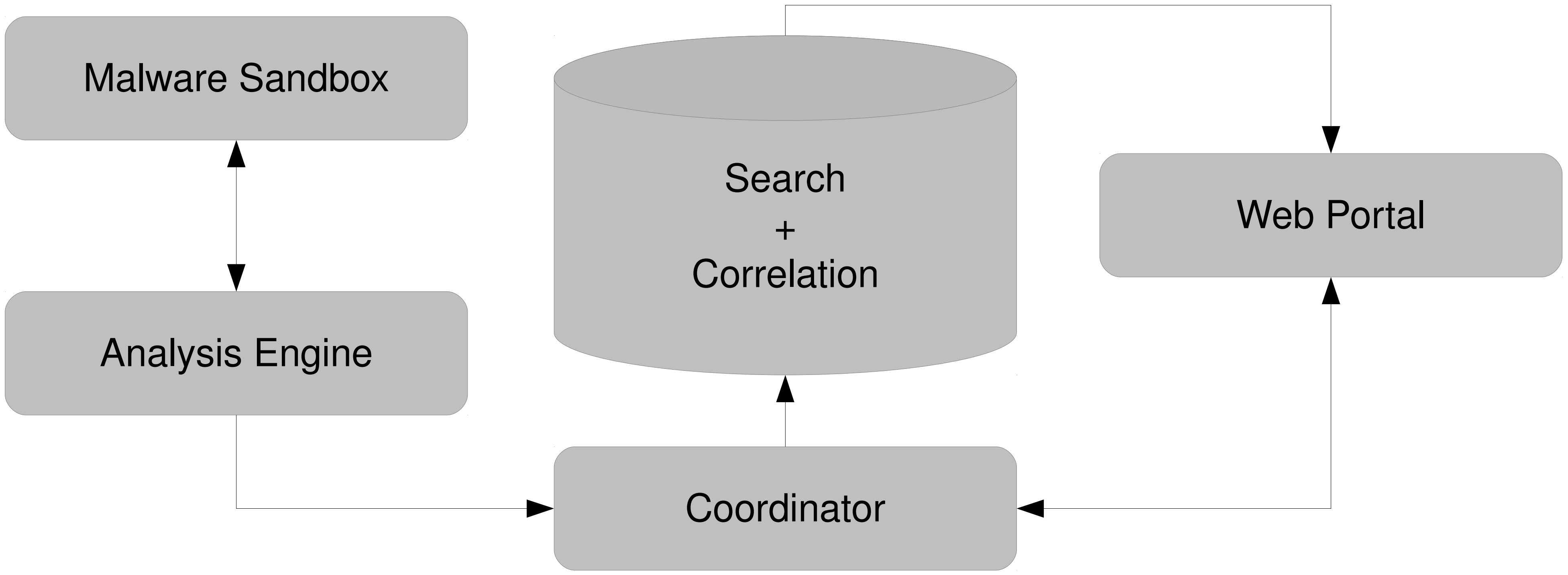
Composing Queries

```
(defn merge-ioc-data [existing-results]
  (let [data (group-by :sample
                      (-> (select* :sample_iocs) ...
                          (execute-select)))]
    (map
     (fn [r]
       (assoc r :iocs (get data (:sample r))))
     existing-results)))
```

ThreatGRID

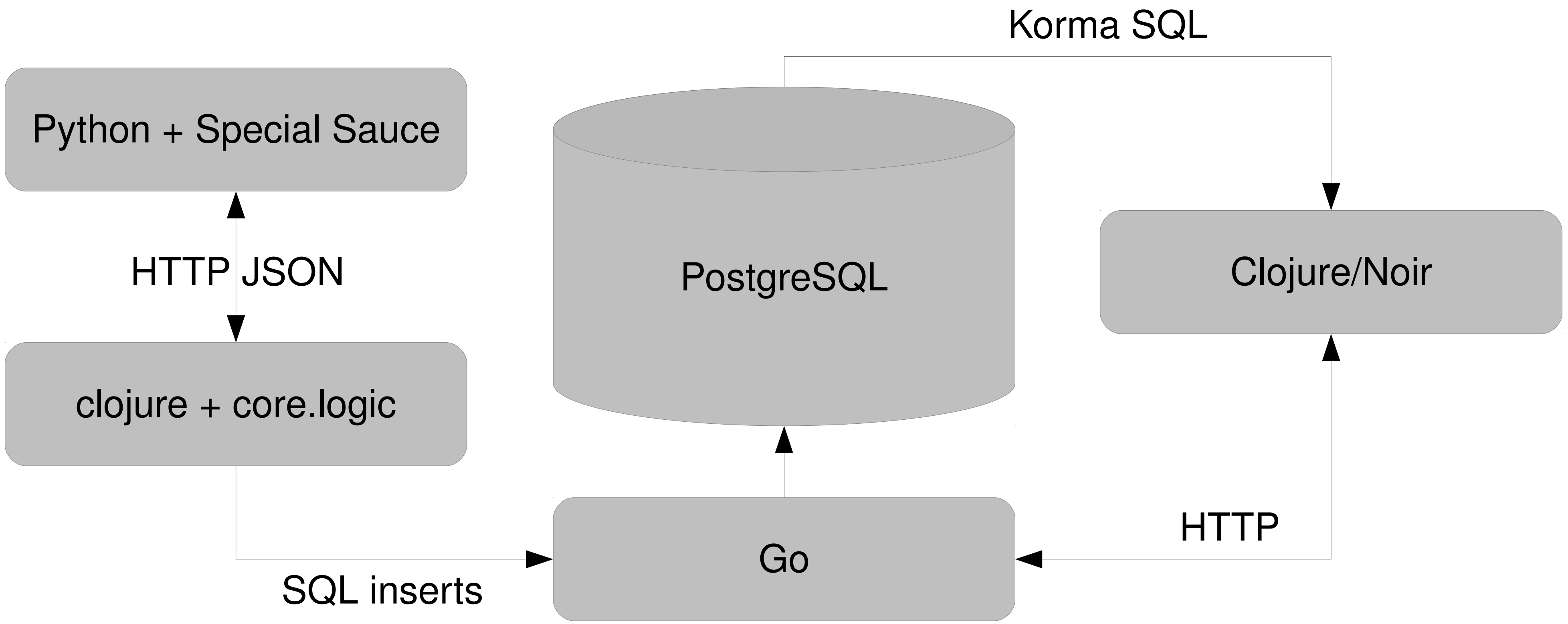
Malware Analysis & Threat Intelligence

The Vision

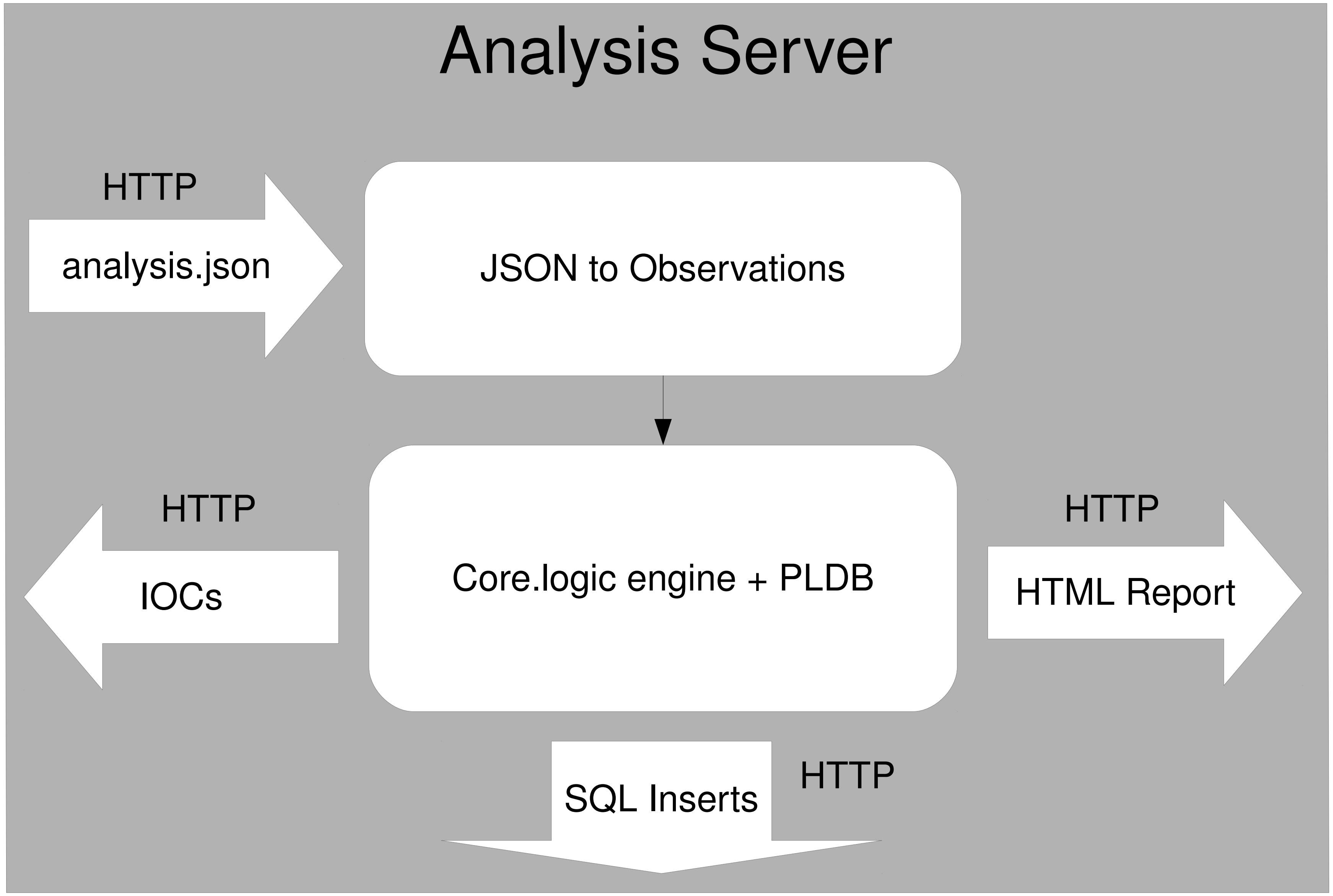


ThreatGRID
Malware Analysis & Threat Intelligence

The Vision



Malware Analysis Engine



Data Model Wins

- Relational data model across whole system
- Documentation generated from Observations
- Flexible, we can learn what questions to ask
- Robust in face of malware

Scaling Import

- No artificial keys means no read import
- No multi-table locks, no transaction conflicts
- One importer per table (in Go)

Clojure Web Portal

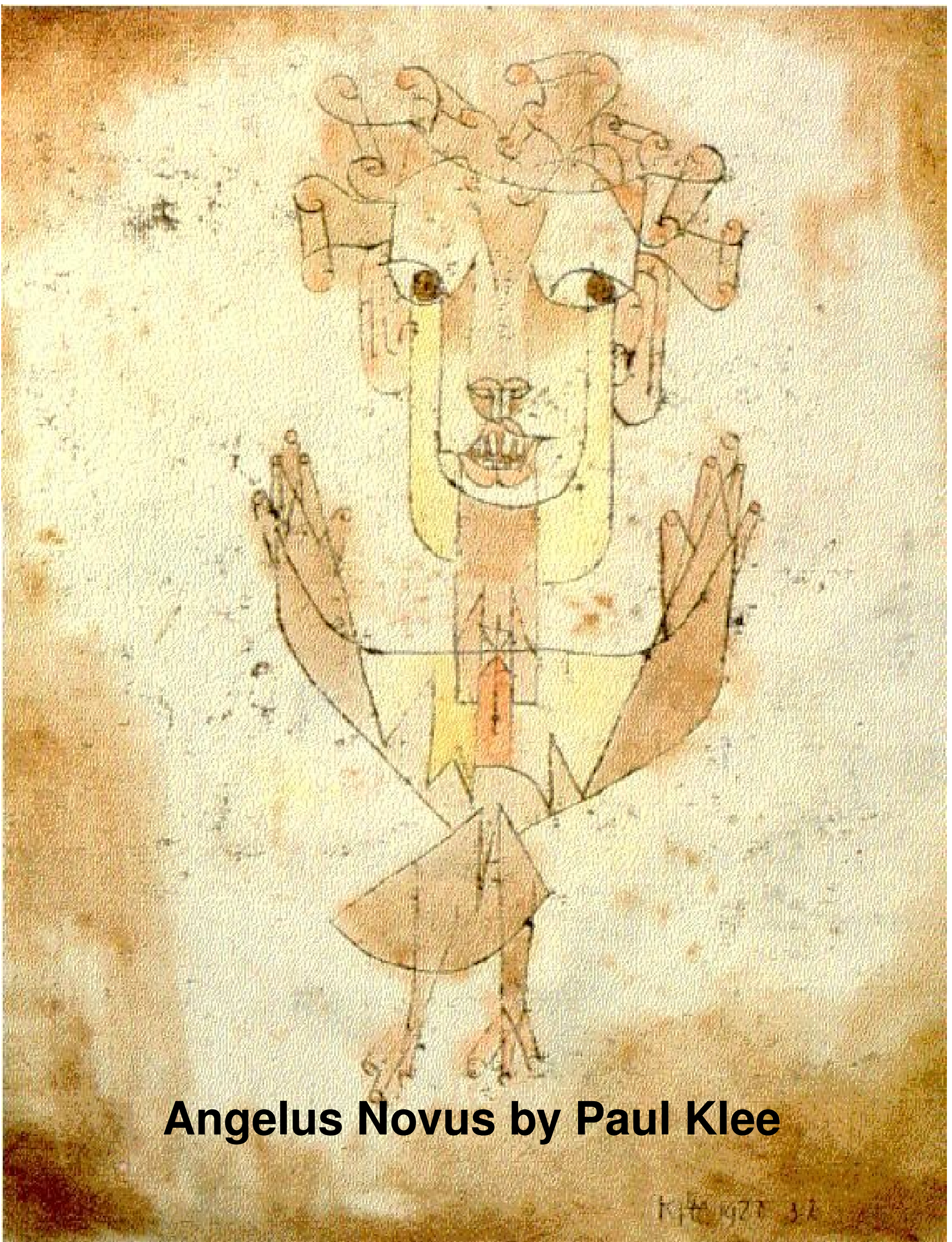
- Sample reports pregenerated
- Faster and more sophisticated queries
- Obvious JVM wins
- Built on top of HTTP API

It works, man, it works

- Supports our malware analysts
- Imports ~1k samples an hour
- Processing ~50 simultaneous samples in a single process, ~4k an hour.
- Deployed as an appliance

Takeaways

- relational model needs no abstraction on top
- stay close to your data
- core.logic lets you build powerful systems
- Balance the concrete and the abstract
- http://github.com/threatgrid/observations


Angelus Novus by Paul Klee

The storm irresistibly propels him into the future to which his back is turned, while the pile of debris before him grows skyward. This storm is what we call progress.

- Walter Benjamin