HS SEDI
Homeland Security Systems Engineering and Development Institute

Penny Chase

Ivan Kirillov – Desiree Beck – Robert Martin

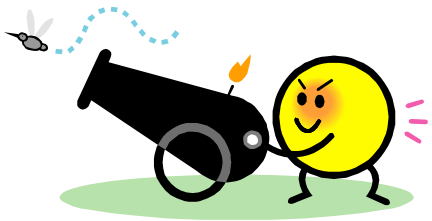
Homeland Security

Why Do We Need to Develop Standards for Malware?

Multiple layers of protection

Lots of products

Inconsistent reports



There's an arms race

Correlate, Integrate, Automate

Threats

MAEC™

Detection

Vulnerabilities

Platforms

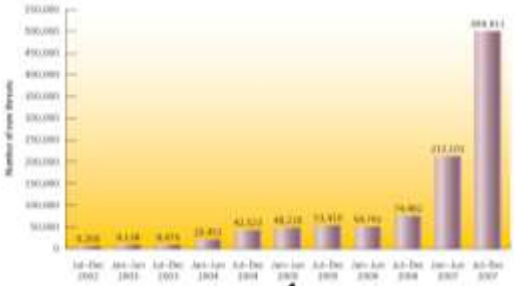
Response

Background

HS SEDI
Homeland Security Systems Engineering and Development Institute

Rise of New Threats

Symantec Global Internet Security Threat Report, Volume XIII, 4/2008



Nimda or I-Worm or Readme?

Oct 2004

Initial CME discussions at VB Conference

Feb 2005

CME Submission Server

Oct 2005

CME public announcement and website

Jan 2007

39 CME IDs assigned

Feb 2007

DHS SwA Forum Malware WG

Dec 2009

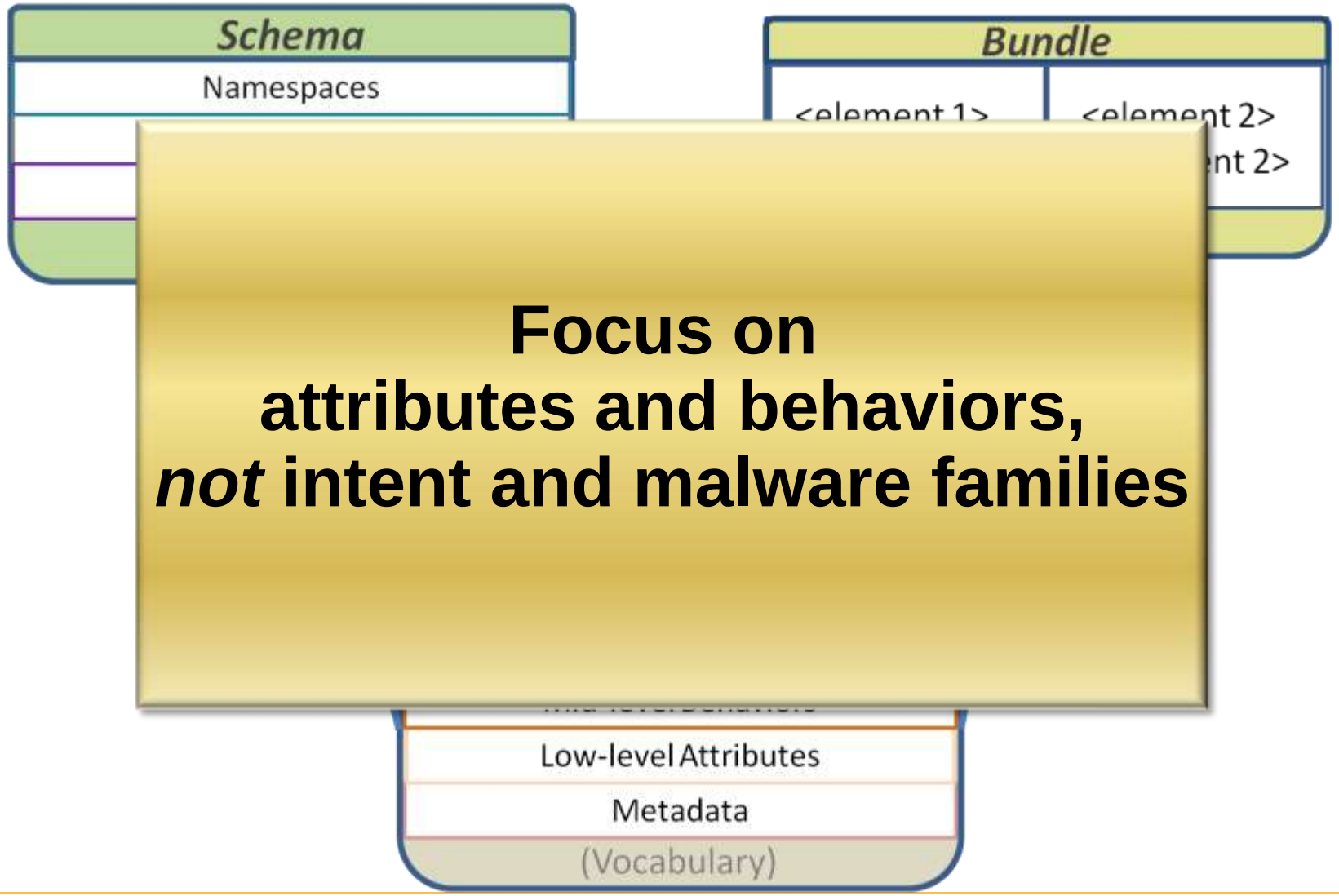
MAEC public website

Jun 2010

Initial MAEC Schema

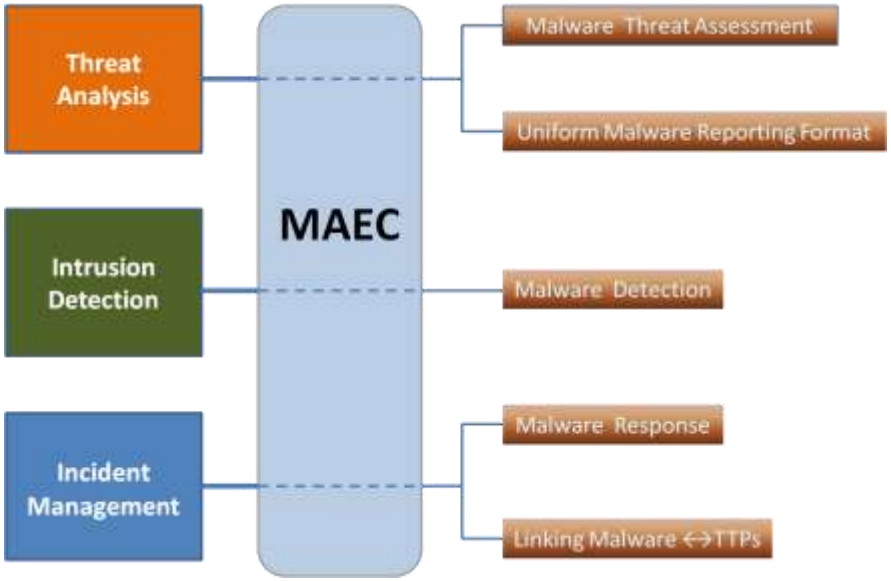
CME → MAEC™

Malware Attribute Enumeration and Characterization (MAEC)

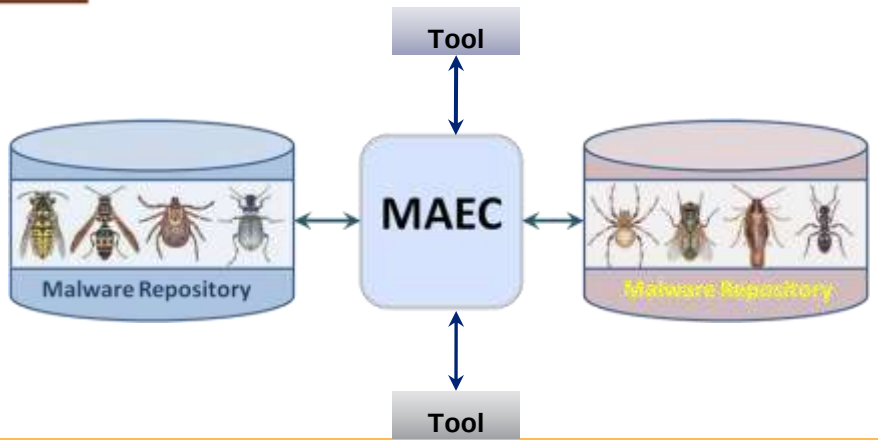


MAEC Use Cases

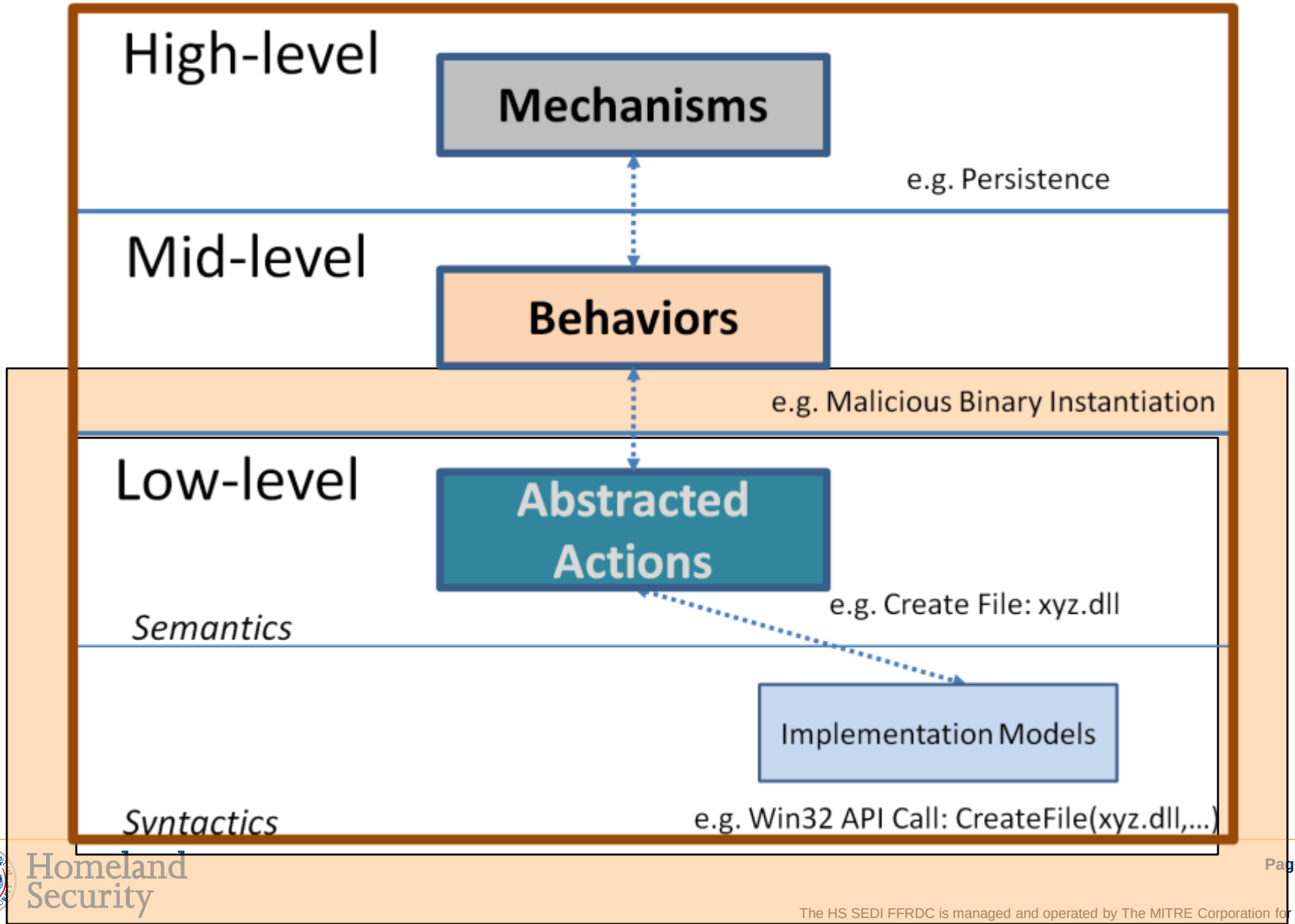
- **Operational**



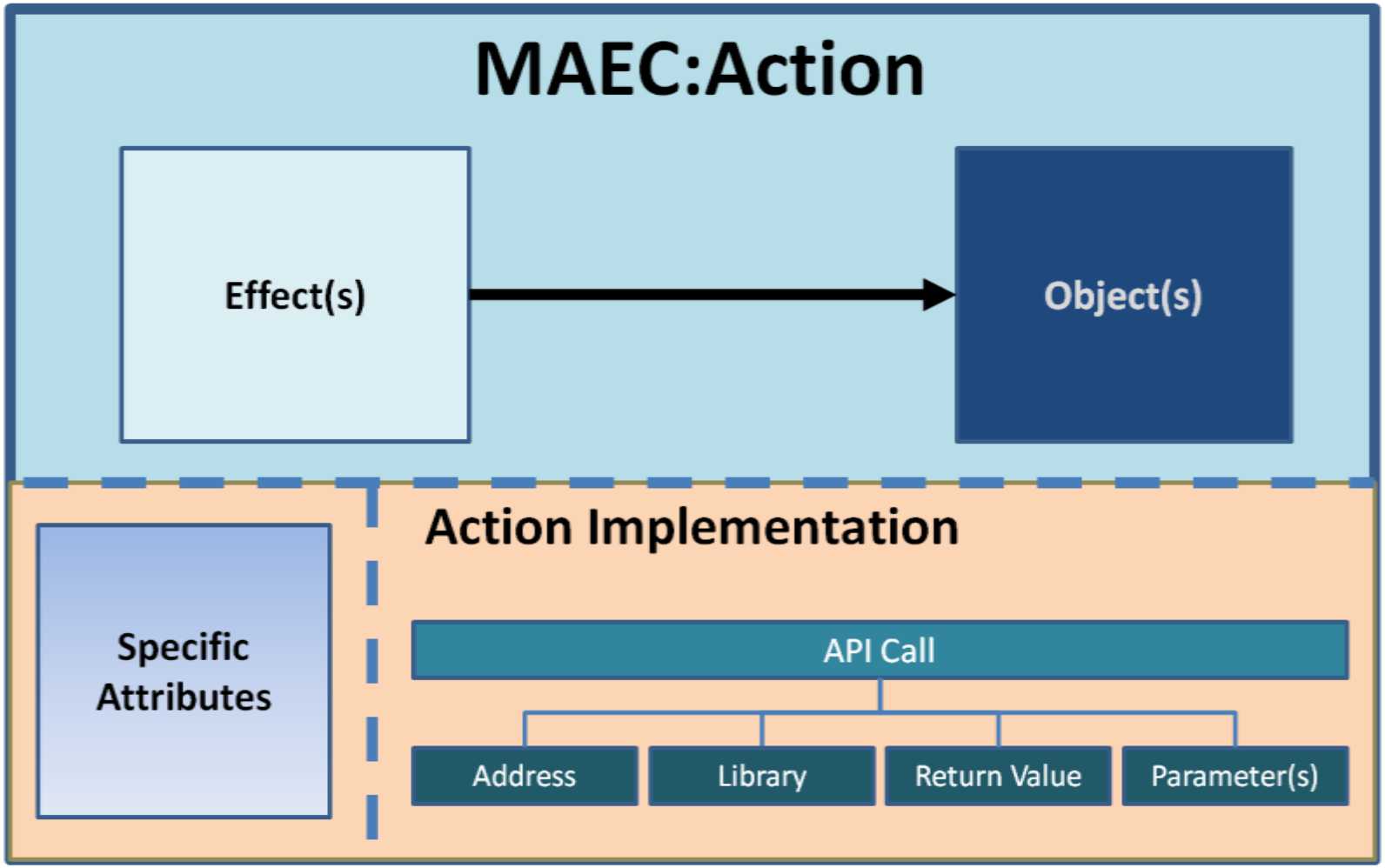
- **Analysis**
  - Help Guide Analysis Process
  - Standardized Tool Output
  - Malware Repositories



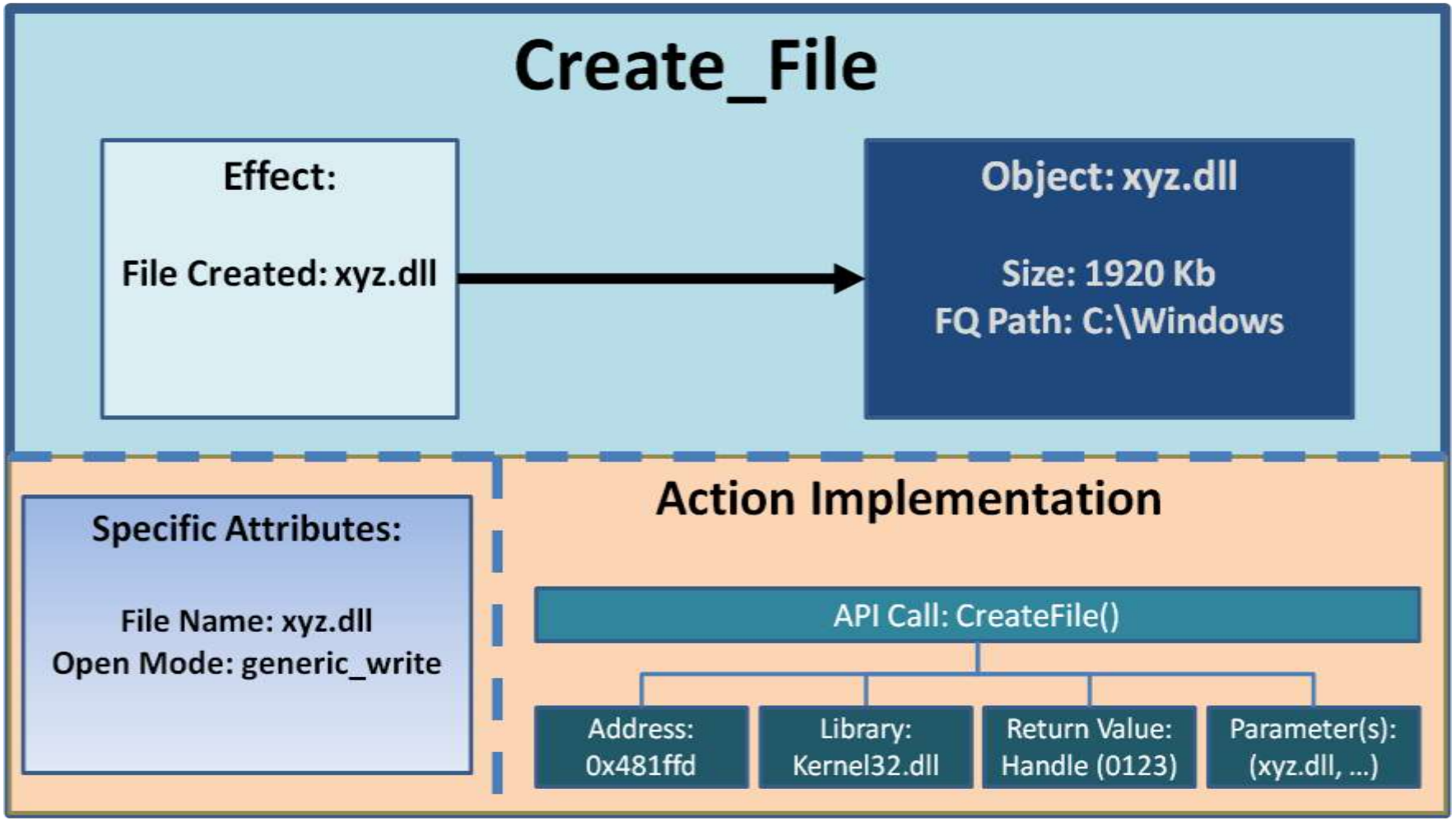
MAEC Overview



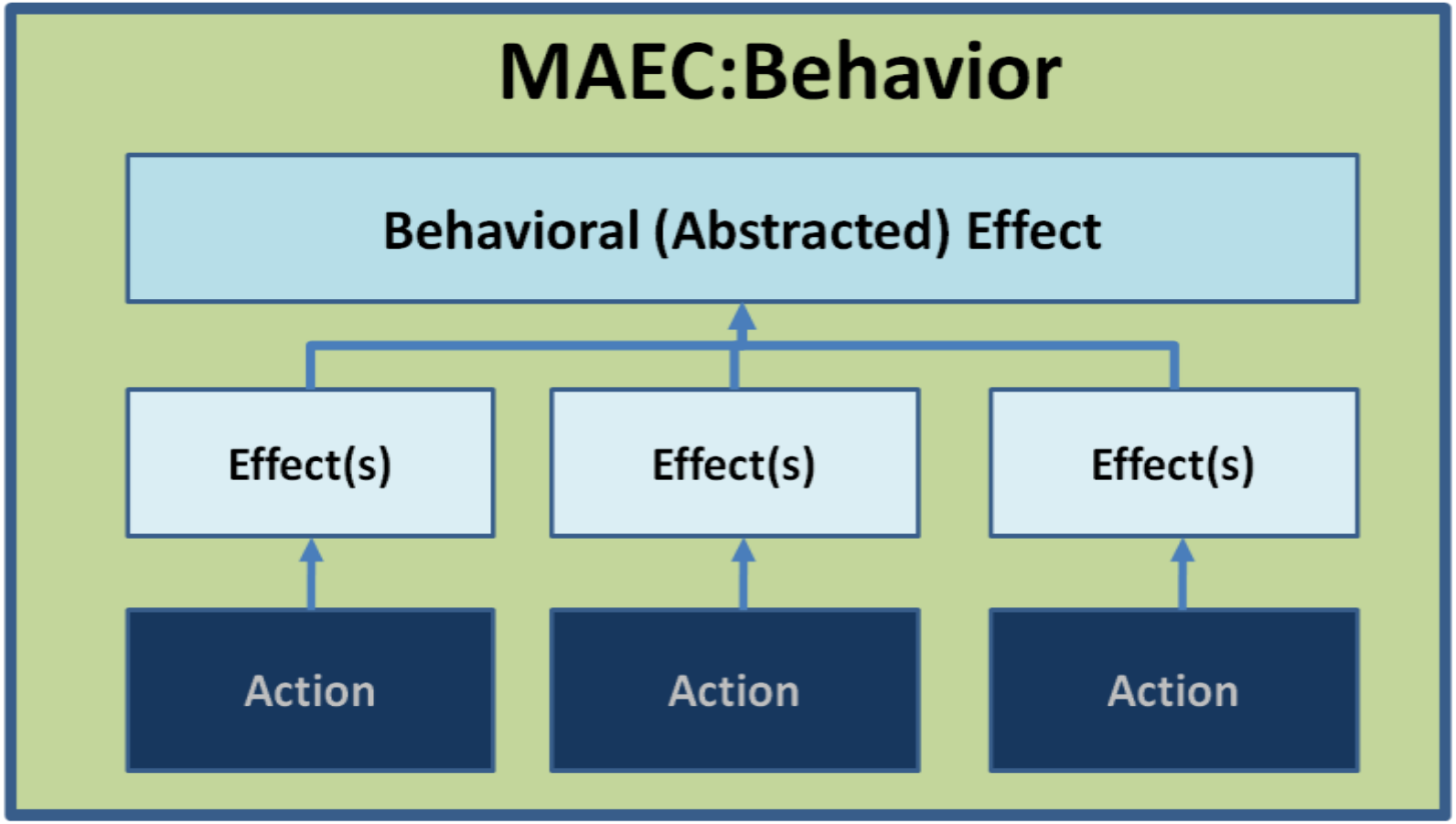
MAEC Action Model



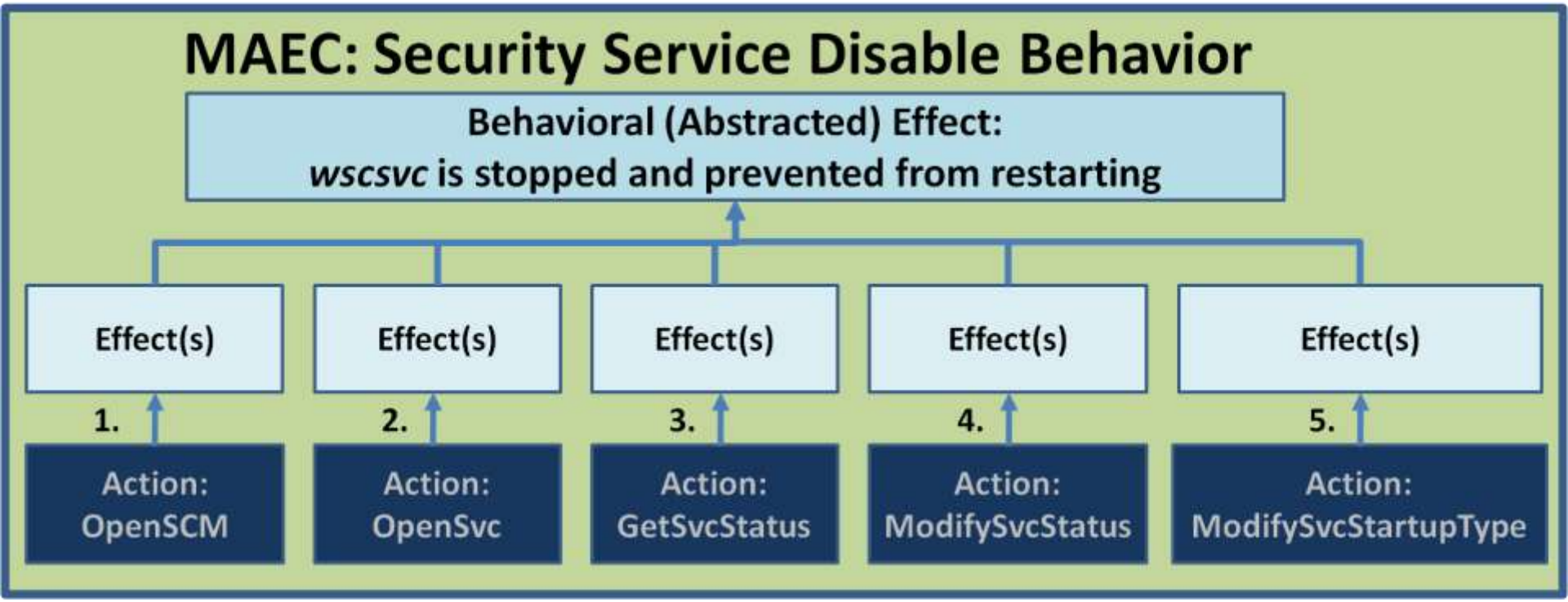
Action Example



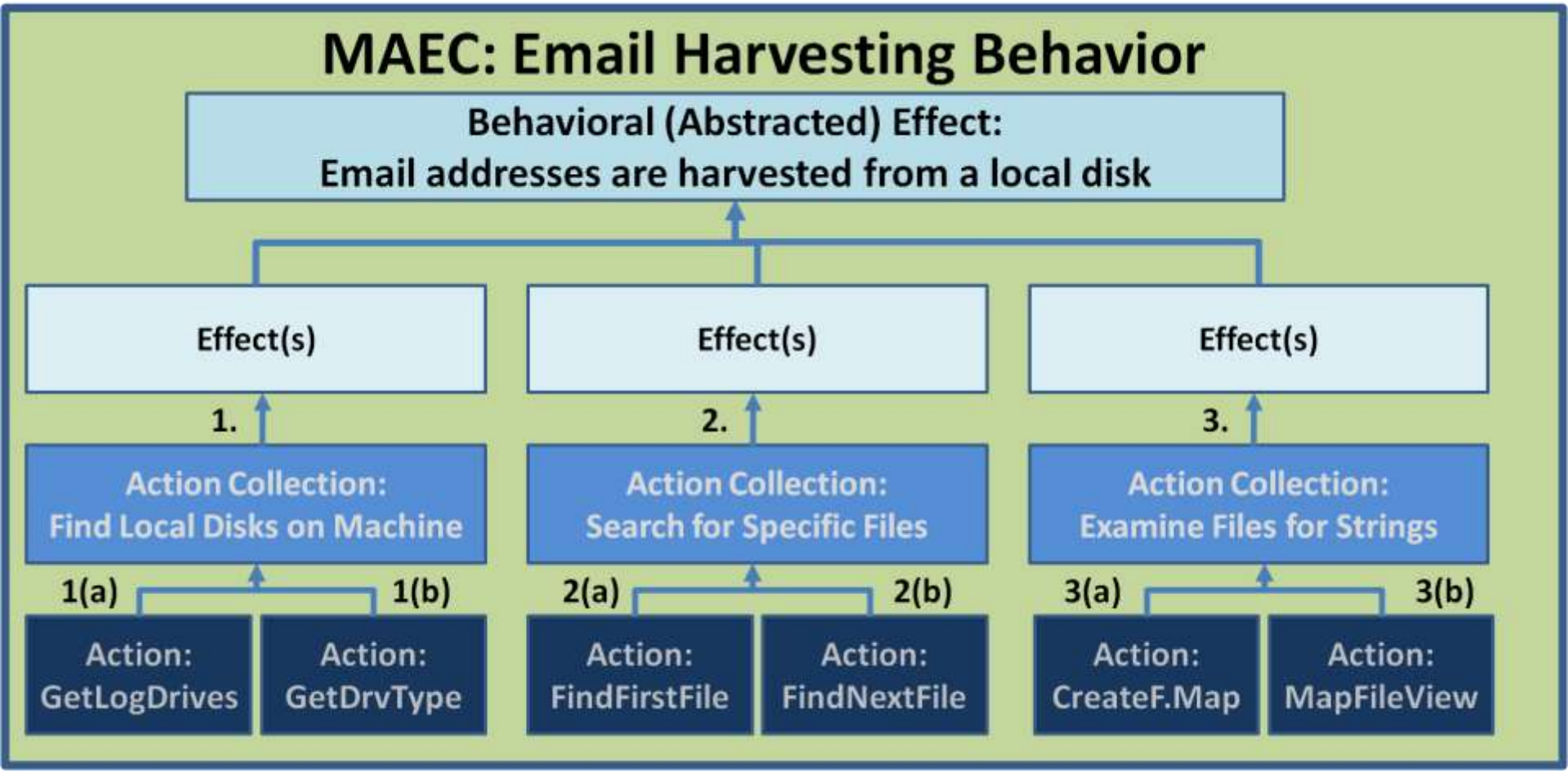
MAEC Behavior Model



Basic Behavior Example



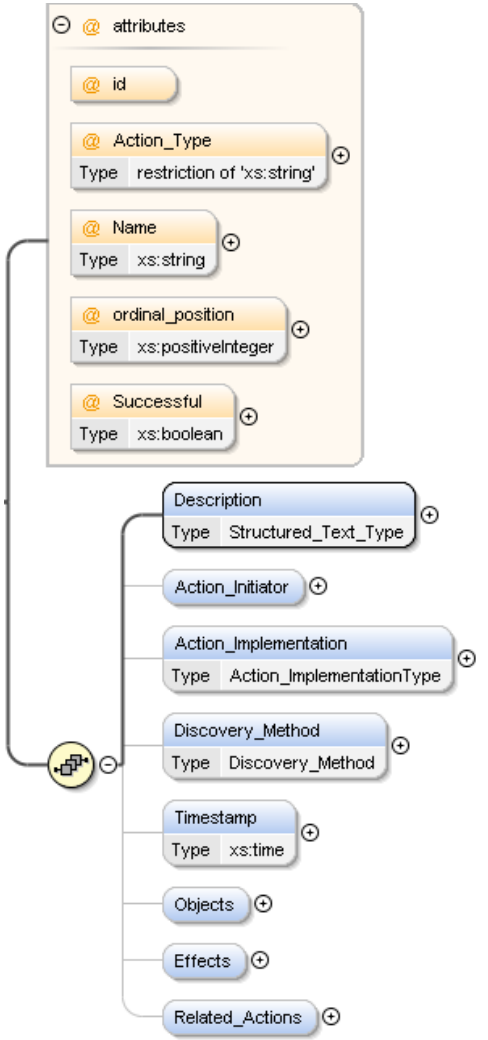
More Complex Behavior Example



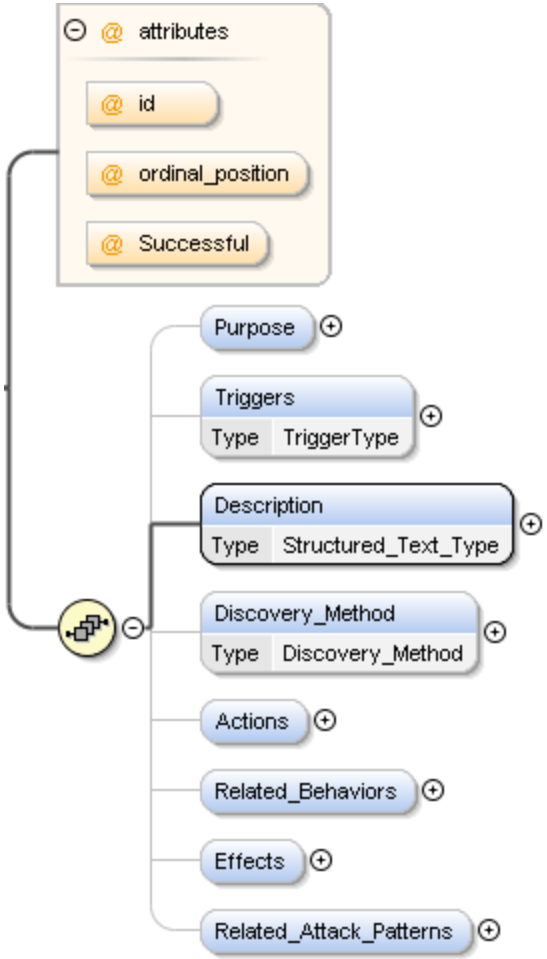
MAEC Schema Overview – Initial Release

HS SEDI
Homeland Security Systems Engineering and Development Institute

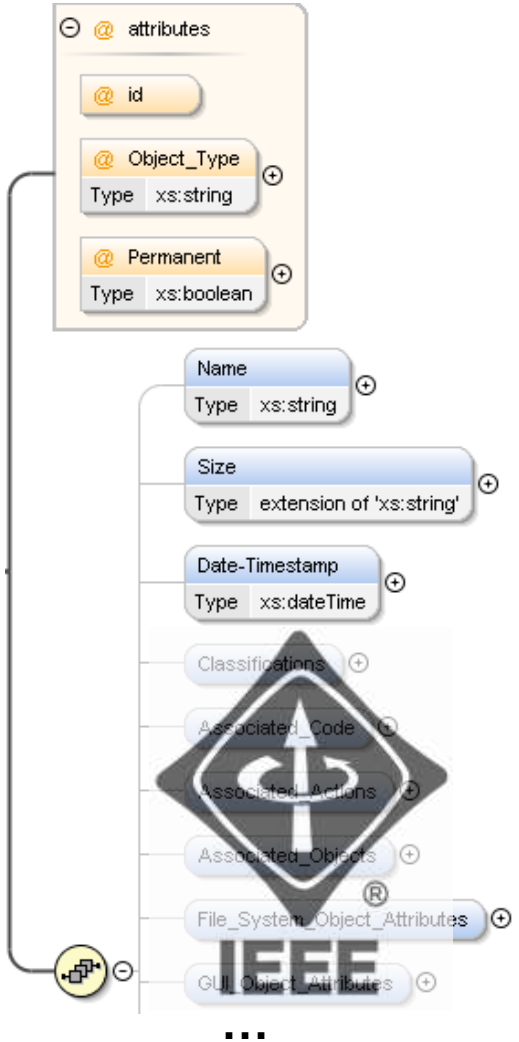
ActionType



BehaviorType



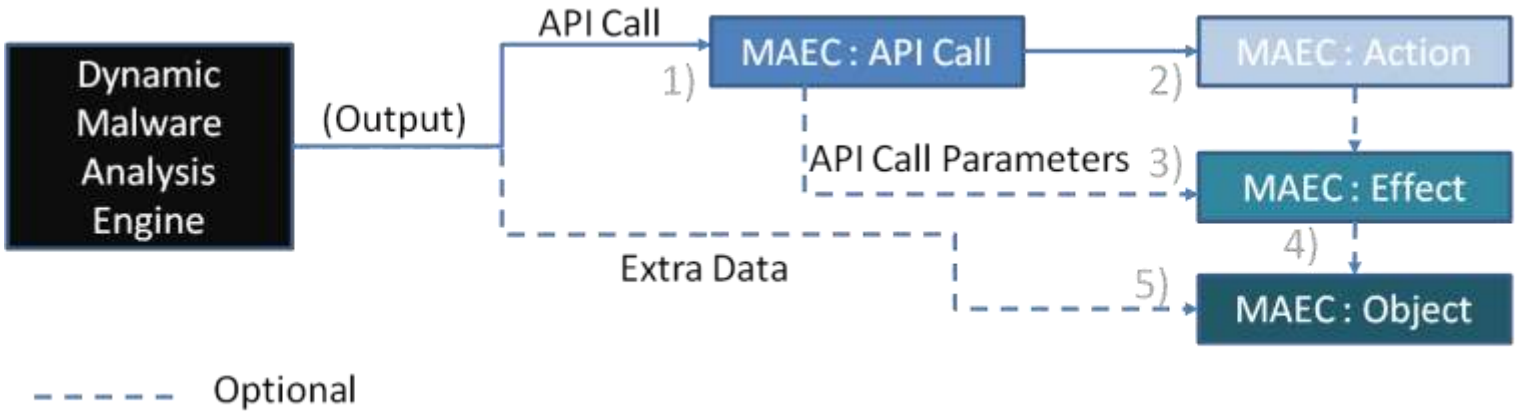
ObjectType



...

Homeland Security

Dynamic Malware Analysis <-> MAEC



Process

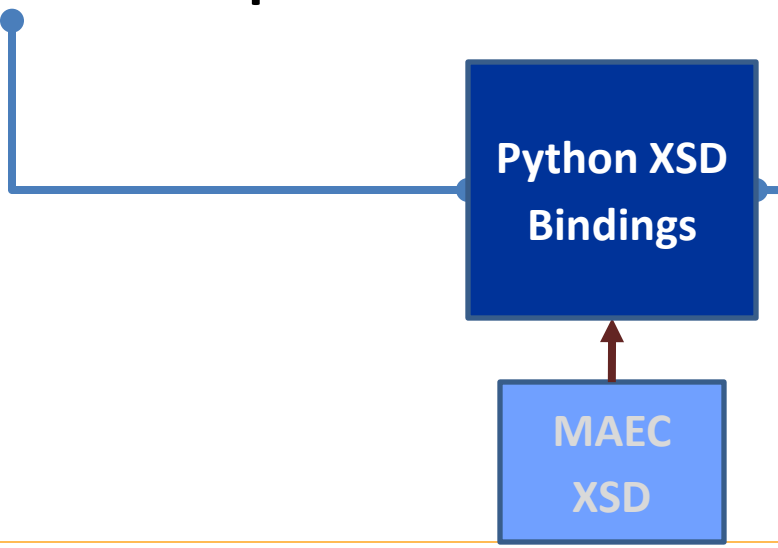
1) An API call is captured by the analysis engine and mapped to MAEC's enumeration of API calls.
2) The MAEC enumerated call is mapped to its corresponding action.
3) The MAEC defined action is mapped to a corresponding MAEC effect (as necessary), which is populated by the parameters of the call.
4) The MAEC effect is linked to a MAEC object (as necessary).
5) Any extra data output (e.g. file attributes, network capture, etc.) from the analysis engine is mapped to its corresponding object (as necessary).

Test Case: CWSandbox Output -> MAEC

HS SEDI
Homeland Security Systems Engineering and Development Institute

```
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."FindFirstFile
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."SetFileAttrib
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."DeleteFileW" -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumKeyA" - <e
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumKeyA" - <e
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumValueW" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumValueW" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExW" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExW" - -
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegCreateKeyExW" -
```

Raw CWSandbox Output




```
<Action Successful="true" id="10" Action_Type="copy" Name="copy_file">
    <Description/>
    <Action_Initiator type="Process">
        <Initiator_Name>KB823988.exe</Initiator_Name>
        <Process_ID>1080</Process_ID>
        <Thread_ID>1812</Thread_ID>
    </Action_Initiator>
    <Action_Implementation>
        <API_Call>
            <Name>CopyFileW</Name>
            <API_Call_Parameter ordinal_position="1">
                <Name>filetype</Name>
                <Value>file</Value>
            </API_Call_Parameter>
            <API_Call_Parameter ordinal_position="2">
                <Name>srcfile</Name>
                <Value>c:\\KB823988.exe</Value>
            </API_Call_Parameter>
            <API_Call_Parameter ordinal_position="3">
                <Name>dstfile</Name>
                <Value>C:\\WINDOWS\\system32\\ntos.exe</Value>
            </API_Call_Parameter>
            <API_Call_Parameter ordinal_position="4">
                <Name>creationdistribution</Name>
                <Value>CREATE_ALWAYS</Value>
            </API_Call_Parameter>
            <API_Call_Parameter ordinal_position="5">
                <Name>desiredaccess</Name>
                <Value>FILE_ANY_ACCESS</Value>
            </API_Call_Parameter>
            <API_Call_Parameter ordinal_position="6">
                <Name>flags</Name>
                <Value>SECURITY_ANONYMOUS</Value>
            </API_Call_Parameter>
        </API_Call>
    </Action_Implementation>
```

MAEC XML

- MAEC Actions
- MAEC Objects
- MAEC Behaviors

Sandbox → MAEC Translator Overview

- **Intended as a proof of concept for MAEC**
- **Currently implemented:**
  - 

    http://www.sunbeltsandbox.com
  - Sandnet/Vigilant (MITRE developed)*
    *Not a translator - supports direct output of MAEC XML
- **In development:**
  - Anubis
    http://anubis.iseclab.org
  - 

    http://www.threatexpert.com

Other Work: MAEC → OVAL

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **MAEC XML to OVAL XML Converter**
  - Extracts MAEC Objects (defined as being created by malware)
  - Converts Objects into OVAL Representations
  - Creates definitions and tests to check for the existence of these objects
- **Capabilities/Use cases**
  - When used with an OVAL interpreter, it permits the automated testing of the existence of malware artifacts on any host system
  - Facilitates the interconnection of malware analysis and malware response
- **Currently supported artifacts:**
  - (Windows) Files/Directories/Named Pipes
  - Registry Keys

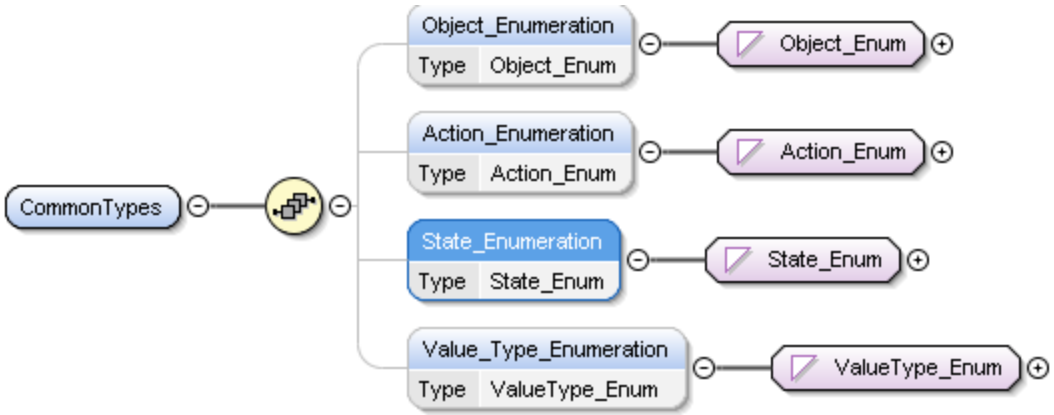
Ongoing Collaboration

- **IEEE ICSG Malware Working Group**
  - Developed Malware Metadata exchange schema to facilitate the sharing of sample data between AV product vendors
    - Attributes for AV classifications, source (URIs), object properties (file hashes, registry keys), boolean properties (isKernel, isPolymorphic)
  - MAEC currently imports the IEEE ICSG Malware Metadata exchange schema
  - In the future, the IEEE schema may import certain MAEC elements

- **Industry /Government**
  - Although non-standardized, there has been some related work in this realm done by industry and government
  - We are actively collaborating with several companies on how to best leverage each other's efforts
  - Likewise, we are planning on leveraging the work done by government in the anti-malware space

Emerging Collaboration

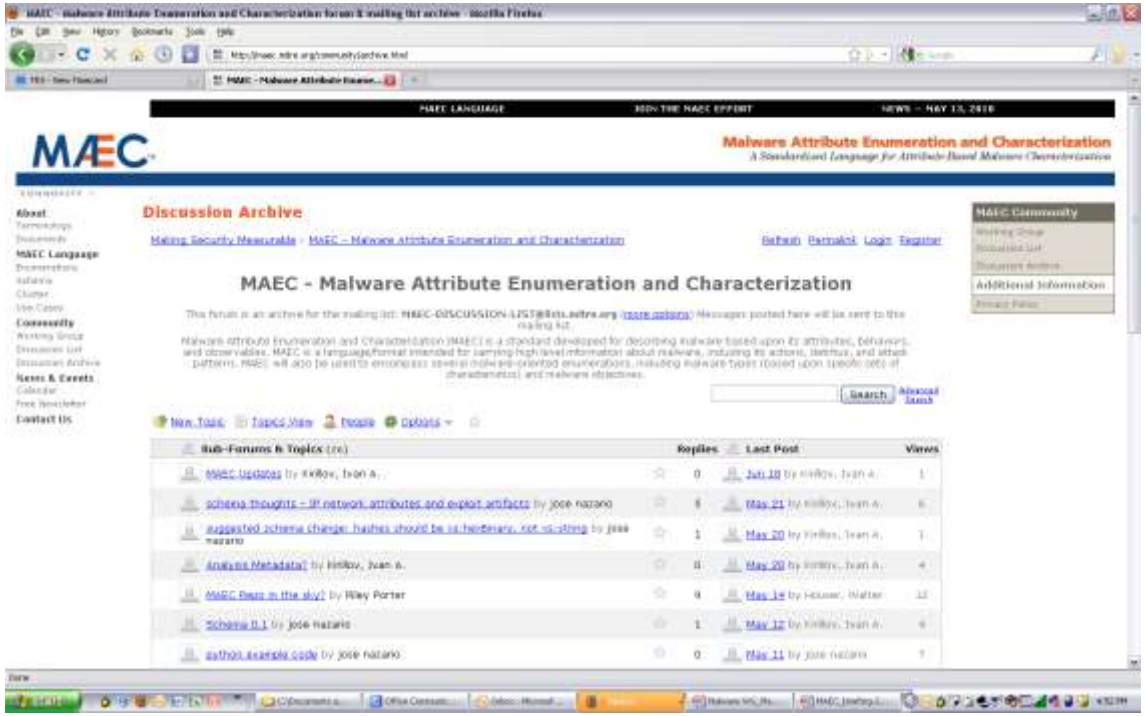
- **Related MSM Efforts**
  - There is significant overlap between MAEC, CAPEC, and CEE in describing observed actions, objects, and states.
  - As such, we're working on developing a common schematic structure of observables for use in these efforts:



- **Others**
  - Feature requests on Handshake group, discussion list
    - Anubis & ThreatExpert translators are being developed as a result of a user request
    - We encourage submission of any other such requests

MAEC Community: Discussion List

- **Request to join: http://maec.mitre.org/community/discussionlist.html**
- **Archives available**



MAEC Community: MAEC Development Group on Handshake

- MITRE hosts a social networking collaboration environment: https://handshake.mitre.org
- Supplement to mailing list to facilitate collaborative schema development

Current Status

- **Initial Schema Release**
  - V1.01 – intended to cover host-based attributes obtained through dynamic analysis/sandboxes
  - Soon to be released on public website
  - Available immediately on Handshake group
- **Translator Tool Development Ongoing**
  - CWSandbox Translator released
  - MAEC -> OVAL converter released
  - Anubis, ThreatExpert translators forthcoming
  - All tools are available on Handshake group

Future Development Plans

- **Expand MAEC coverage of network attributes**
  - Possible focus: bots/botnets
- **Create RDF/OWL ontology based on MAEC schema**
- **Revise schema to better support characterization of relationships between actions/behaviors**
- **Implement common observables schema**
  - Based on MAEC/CAPEC/CEE collaboration
- **Encourage and invite more participation in the development process**
  - MAEC Website: http://maec.mitre.org (contains MAEC Discussion list sign-up)
  - MAEC Handshake Group

Summary

- MAEC is attempting to address many of the issues that are integral to accurate and unambiguous communication about malware
- The adoption of MAEC will facilitate new methods of correlation and automation against malware
- MAEC is an open, collaborative effort. It needs expertise and input from various parties in order to be successful