Detection as Code;

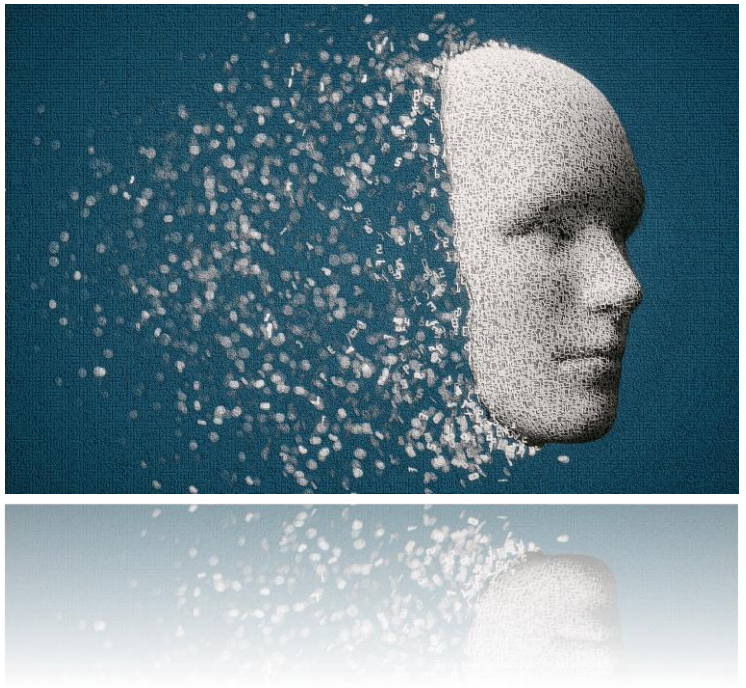
A Sigma Story

28 / 06 / 2024

Universiteit Leiden

Bij ons leer je de wereld kennen

Me



- *MSc. Roy Kokkelkoren*
  - *University of Twente: Computer Security Masters*
  - *7 jaar als senior security specialist bij NCSC-NL*
  - *Sinds eind 2023 SOC-Lead bij Leiden University (LEI)*

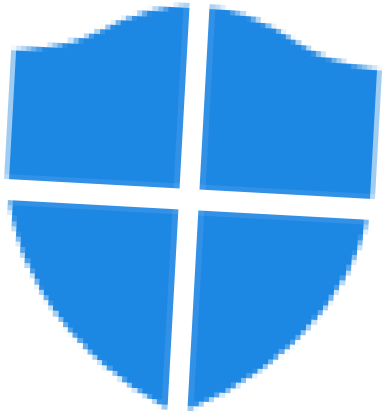
- *Ontwikkeldoel SOC*
  - *Volwassenheidsniveau van SOC-LEI verbeteren*
    - *Standaardiseren Security operations (identification / containment & recovery)*
    - *Verbeteren detection rate*
    - *Verbeteren response time*

1. Probleemdefinitie

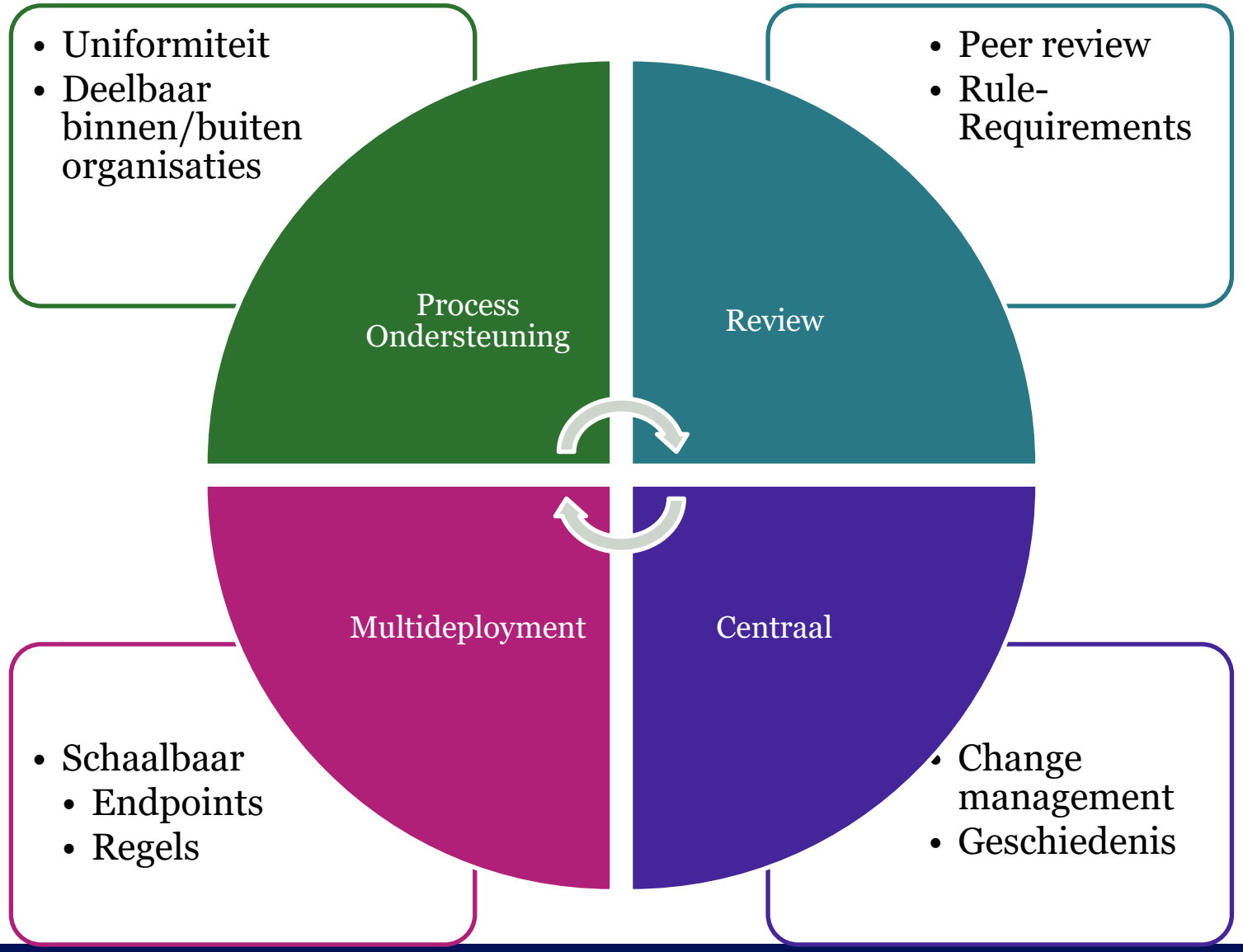
Universiteit Leiden

Bij ons leer je de wereld kennen

Wat is het probleem?



Detection as Code



"Detection as Code is a new paradigm that brings a structured, systematic and flexible methodology for threat detection inspired by the as-code best practices"

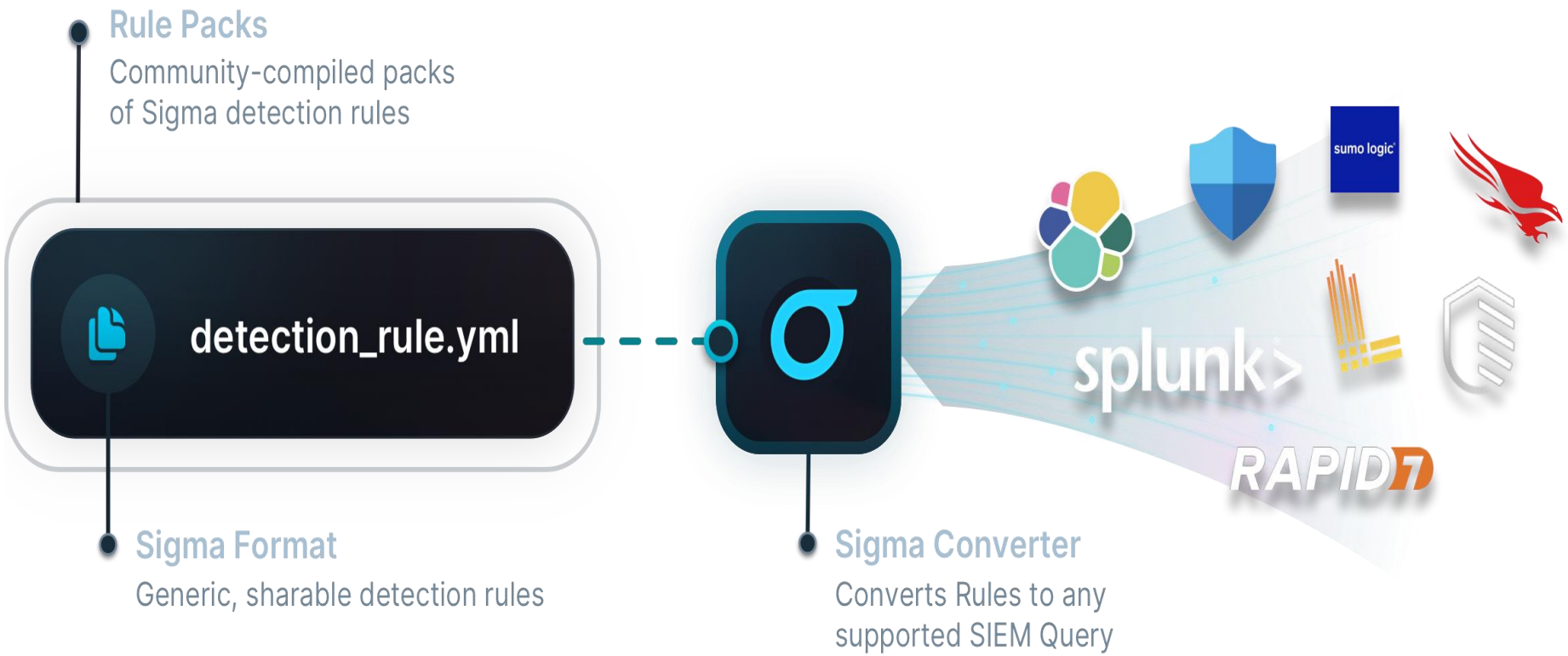
2. Sigma

Universiteit Leiden

Bij ons leer je de wereld kennen

Sigma

- *Ecosystem*
  - *Sigma Format*
  - *CLI-Tools*
  - *VS-code integraties*
  - *pySigma library*
  - *Sigma Rule collective*
- *Producten / Integraties*
  - *Siems*
  - *Sandboxes*
  - *Online diensten (VT)*
  - *Stream analyzers (Kafka)*
  - *XDR*



"Sigma is for logs what Snort is for network traffic and YARA is for files."

Sigma; Ondersteuning

Stable	Stable	Stable	Development	Testing
CarbonBlack	InsightIDR	SentinelOne	Azure Log Analytics Socprime	Datadog
CortexXDR	Loki	SentinelOne PowerQuery	Panther	Hawk
Dictquery	Microsoft 365 Defender	Splunk	Stix	PowerShell
Elasticsearch	OpenSearch	Uberagent	Trellix Helix	Sqlite
IBM QRadar AQL	Qradar			

3. Sigma implementatie

Universiteit Leiden

Bij ons leer je de wereld kennen

Sigma, lei-detectieregel.yml

```yaml
title: Fax Service DLL Search Order Hijack
id: 828af599-4c53-4ed2-ba4a-a9f835c434ea
status: experimental
description: The Fax service attempts to load
ualapi.dll, which is non-existent. An attacker can
then (side)load their own malicious DLL using this
service.
references:
  - https://windows-internals.com/faxing-your-way-to-
system/
author: NVISO
date: 2020/05/04
modified: 2024/01/09
tags:
  - attack.persistence
  - attack.defense_evasion
  - attack.t1112
logsource:
  product: windows
  category: image_load
detection:
  selection:
    Image|endswith:
      - fxssvc.exe
    ImageLoaded|endswith:
      - ualapi.dll
  filter:
    ImageLoaded|startswith:
      - C:\Windows\WinSxS\
  condition: selection and not filter

falsepositives:
  - Unlikely
level: high
```

Sigma, resultaat

sigma convert -p pipeline.yml -t {splunk,lucene} -f default lei-detectieregel.yaml

lucene

```
winlog.channel:Microsoft\-Windows\-Sysmon\/Operational
  AND (event.code:7 AND ((process.executable:*fxssvc.exe
  AND file.path:*ualapi.dll) AND (NOT
  file.path:C\:\Windows\WinSxS\*)))
```

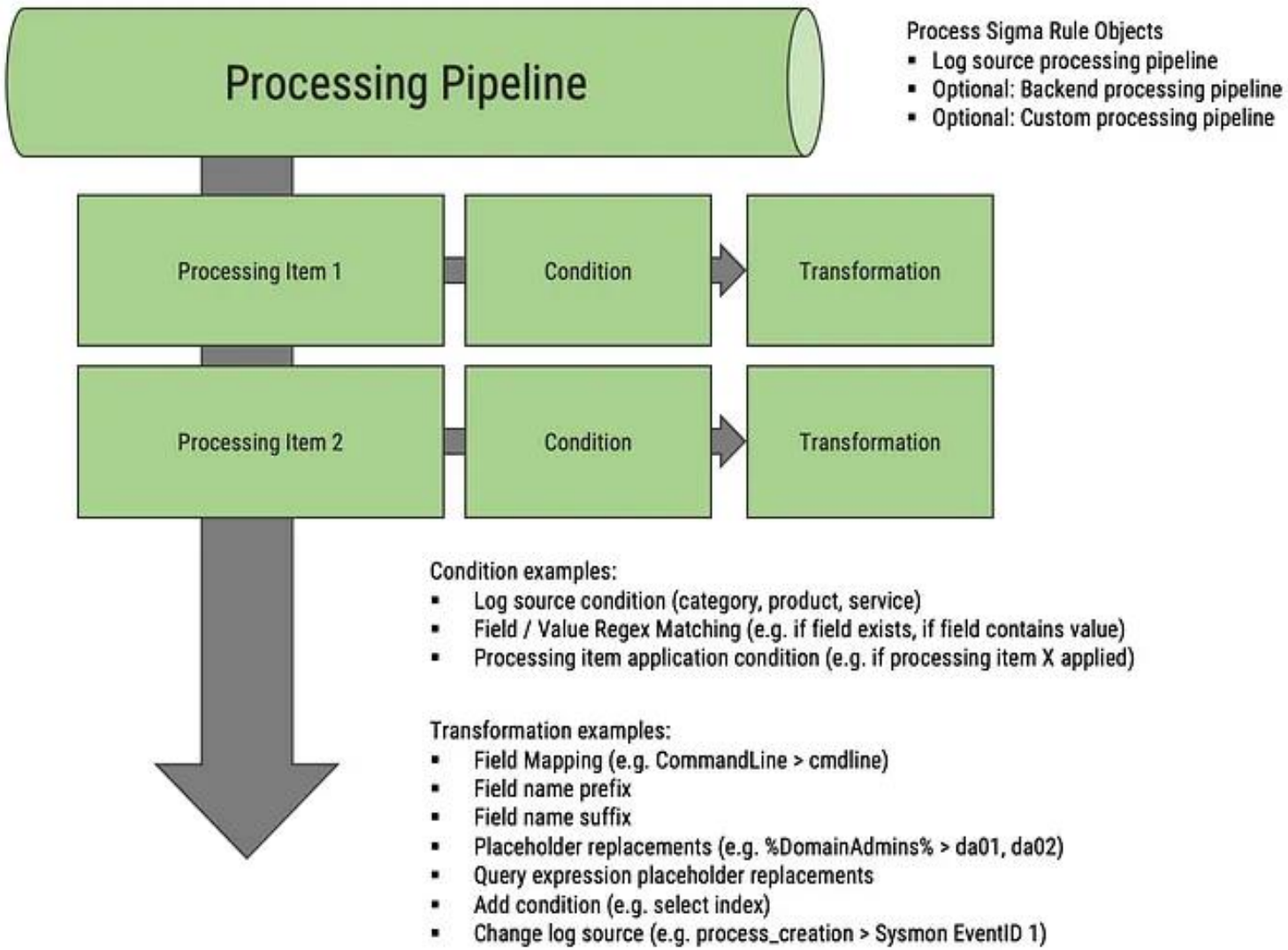
splunk

```
source="WinEventLog:Microsoft-Windows-
  Sysmon/Operational" EventCode=7
  ProcessExecutable="*fxssvc.exe" Path="*ualapi.dll" NOT
  Path="C:\Windows\WinSxS\*"
```

Sigma Pipelines

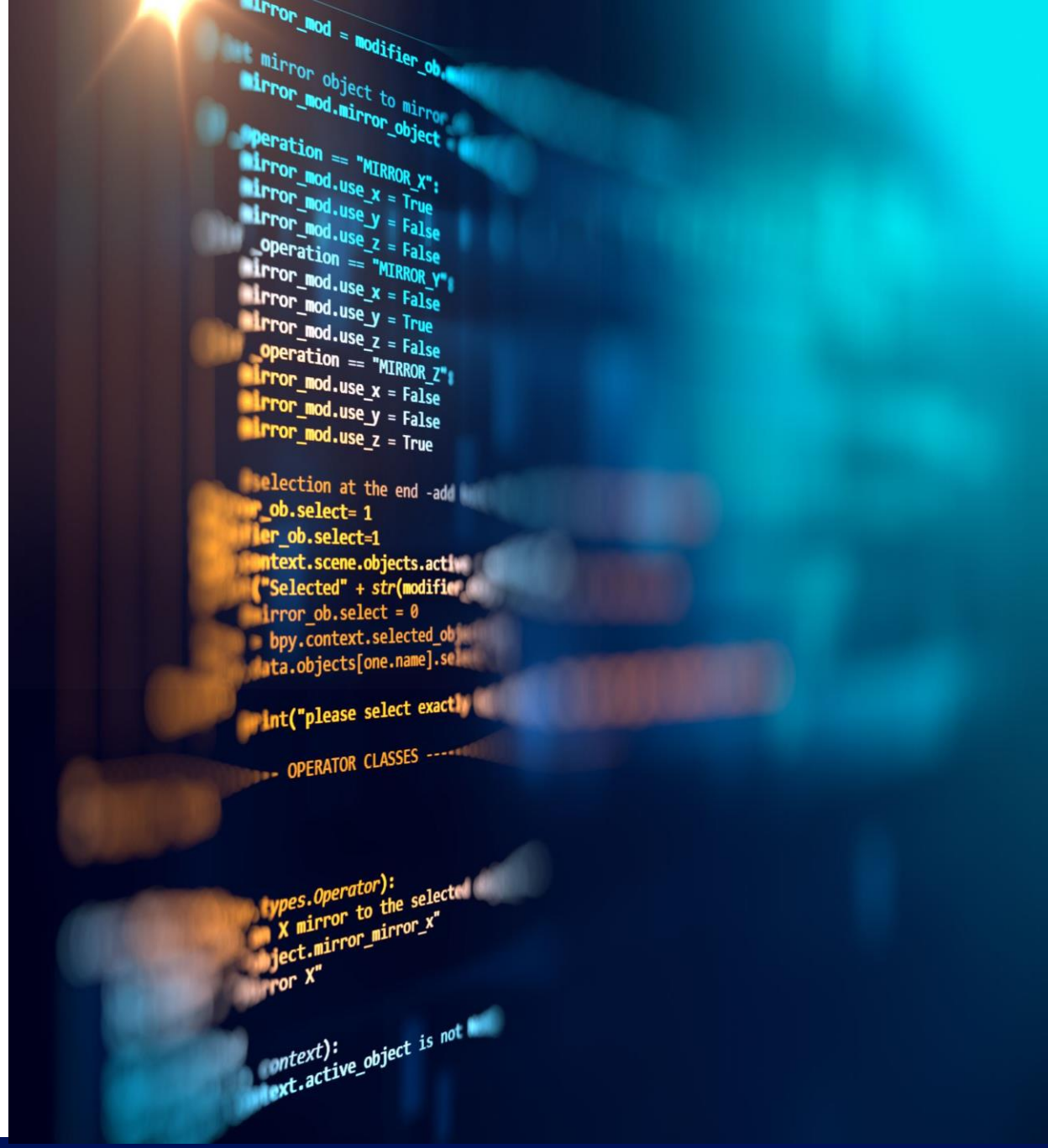
- Pipelines worden gebruikt voor SIEM specifieke vertalingen:
  - Queries uitbreiden
  - Vertalingen veldnamen
  - Pre-/suffixes toevoegen
  - Placeholders vervangen

- Transformaties kunnen gerelateerd worden aan condities
  - Log source validatie
  - Veld / waarde validatie



LEI-detectieregel v2

```
.....
logsource:
    product: m365
    category: activity_logs
detection:
    selection:
        src_ip:
            - 1.1.1.1
    lei_exception_1:
        command:
            - FileDownloaded
            - FileAccessedExtended
        app: OneDrive
    lei_exception_2:
        app:
            - MicrosoftTeams
    condition: selection and not 1 of lei_exception_*
```



Sigma, LEI-detectieregel v2

```
transformation=AddConditionTransformation({
  'index': foo_data_nl_windows',
  'sourcetype': 'o365:management:activity'
}),

transformation=FieldMappingTransformation({
  'src_ip': ['ClientIPAddress', 'OriginAddress'],
}),

rule_conditions=({
LogsourceCondition(product="m365"),
LogsourceCondition(category="activity_logs")
})
```

Splunk query:

```
index="foo_data_nl_windows"
sourcetype="o365:management:activity"

ClientIPAddress="1.1.1.1" OR
OriginAddress="1.1.1.1"

NOT (
   (
    command IN (
      "FileDownloaded",
      "FileAccessedExtended"
    )
    app="OneDrive"
   )
   OR app="MicrosoftTeams"
)
```

4. Optima

Universiteit Leiden

Bij ons leer je de wereld kennen

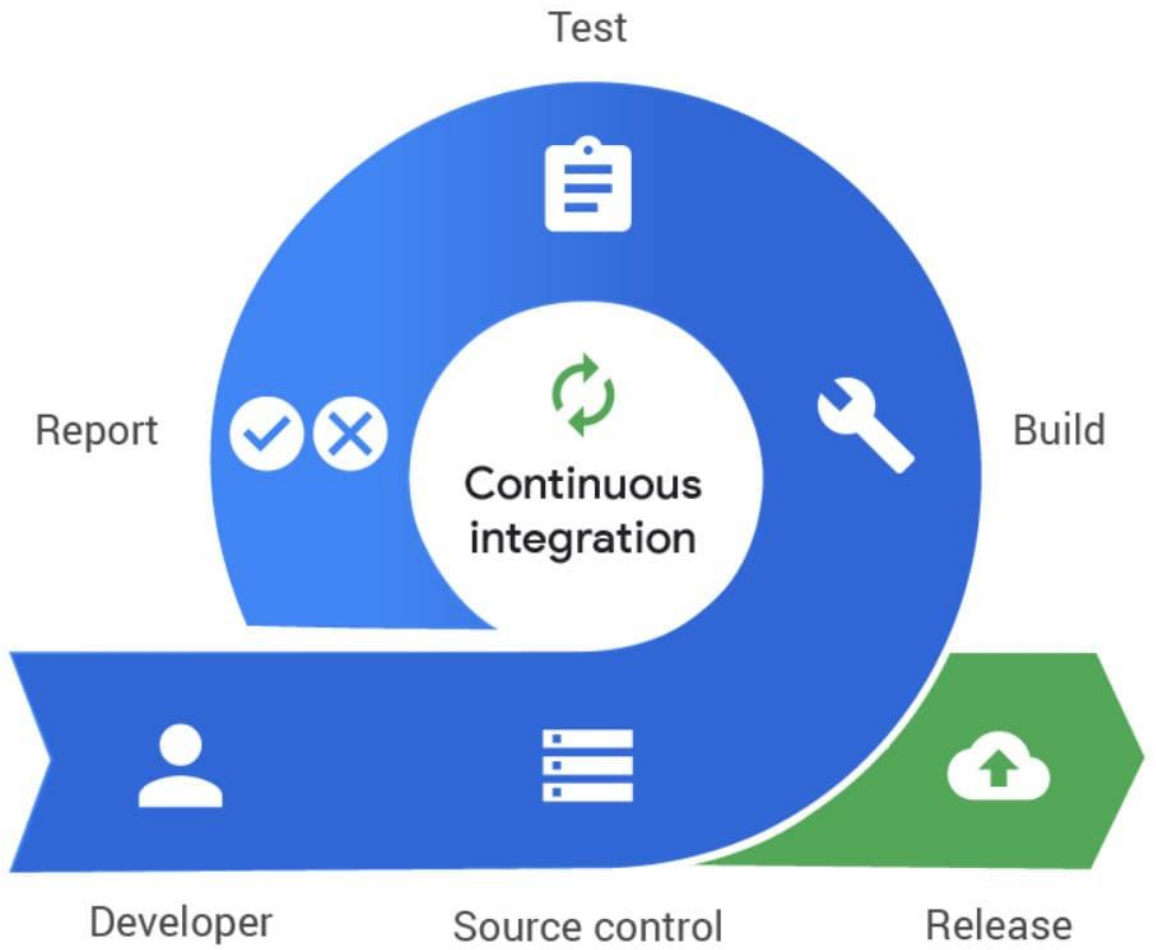
Sigma Optima – CI/CD

- Volledige automatisering via Gitlab CI/CD

- Peer review wordt gerealiseerd via Merge Request (MR) en protected main branch; approval nodig voor wijzigingen!
- Git commits vereisen toelichting van wijziging.

- Optima verwezenlijkt fase:
  - Build
  - Test
  - Release

- Toegang tot SIEMs wordt afgeschermd via Gitlab Secrets – (HashiCorp Vault)



5. Lessons learned

Universiteit Leiden

Bij ons leer je de wereld kennen

SURFsoc NG

- Waarom heeft elke aparte instantie zijn eigen set aan detectieregels.

- Verwachtiging: detectieregels hebben veel overlap tussen instanties.

- Implementatie van centrale detectieregel repo (ala MISP), waar elke instantie kan bijdragen maar ook gratis uit kan nuttigen.
- Enige lokale vereiste is vertaling naar eigen platformen (Optima / Sigma)

DaC; Lessons Learned

- Sigma is fundamenteel hoe wij onze detectieregels verwerken.
- Sigma maakt het wijzigen van honderden regels eenvoudig.
- Sigma is geen click en forgot, pipelines zullen onderhouden moeten blijven.
- Sigma lost het problem van verificatie van regels niet op, tooling integraties zijn wel makkelijker te realiseren.
- Sommige geadvanceerde detecties kunnen nog niet volledige geimplementeerd worden.
  - Machine Learning alerting in Elastic
  - Geavanceerde gecorreleerde alerts

Voor iedereen die geinteresseerd is in onze Sigma / Optima, kunnen we de code vrij mee delen.

- **r.kokkelkoren@issc.leidenuniv.nl**

Vragen

Universiteit Leiden

Bij ons leer je de wereld kennen