A Choreography-Driven Approach to APIs: The OpenDXL Case Study

Leonardo Frittelli @ McAfee, Cordoba, AR
Facundo Maldonado @ McAfee, Cordoba, AR
Hernán Melgratti @ UBA, AR
Emilio Tuosto @ GSSI, IT & UoL, UK

Coordination 2020 15-20 July 2020

Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233.

BEHAPI Behavioural Application Program Interfaces

Prelude

A fairy tale

(pictures from Matteo Garrone's "Pinocchio")

Tell them...

A fairy tale

(pictures from Matteo Garrone's "Pinocchio")

and they'll behave

A fairy tale

(pictures from Matteo Garrone's "Pinocchio")

unless they don't

A fairy tale

(pictures from Matteo Garrone's "Pinocchio")

So, keep an eye on'em

A fairy tale

(pictures from Matteo Garrone's "Pinocchio")

of course, it's for their own good

At a glance

API-based development

- difficult in theory...
- ...and in practice

BehAPI

- Behavioural specification of APIs can help
  - document
  - monitor
- Case study: OpenDXL

Managing expectations

This work

- reports on a collaboration with industry
- uses existing behavioural types
- proposes a methodology
- strives
  - to be easily usable by non-experts
  - to attain practical benefits

Managing expectations

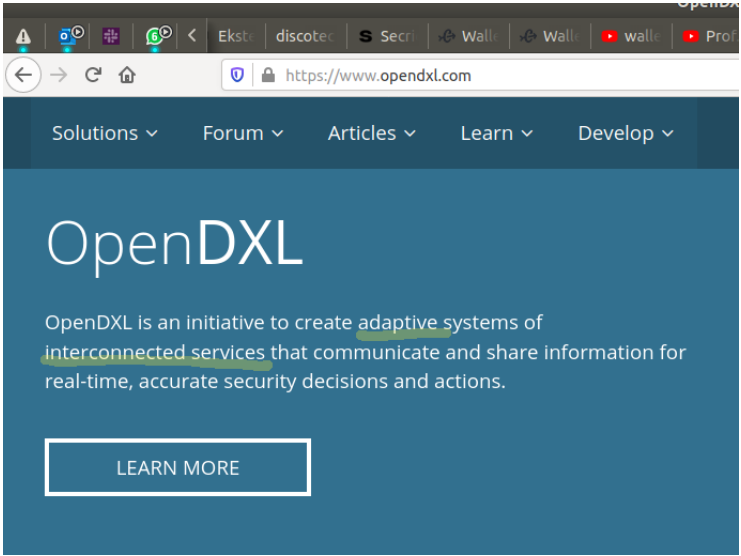
This work

- reports on a collaboration with industry
- uses existing behavioural types
- proposes a methodology
- strives
  - to be easily usable by non-experts
  - to attain practical benefits

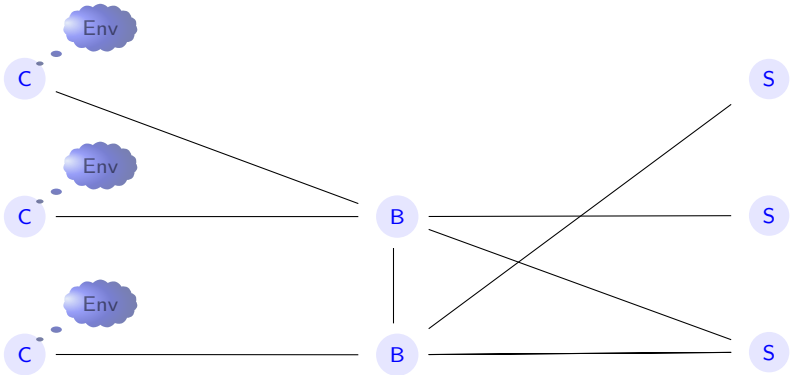
No new technical contributions

OpenDXL
&
Threat Intelligence Exchange

Open Data Exchange Layer

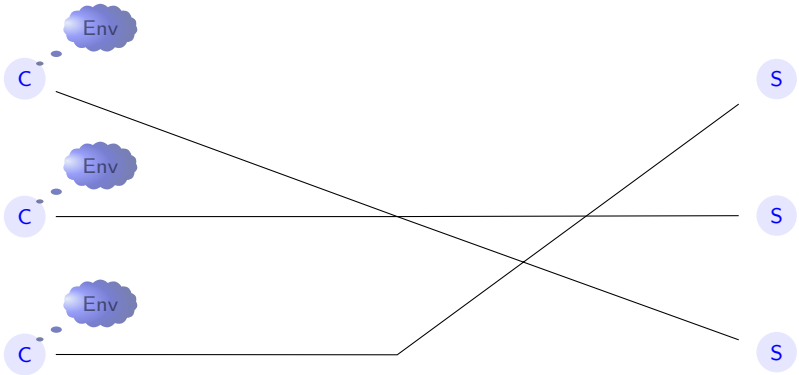


Architecture of OpenDXL



- Brokers abstracted away

Architecture of OpenDXL

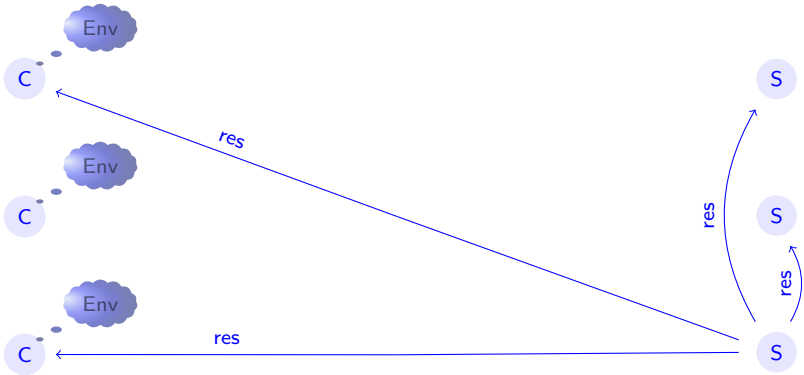


- Brokers abstracted away
- Event-based communication

Architecture of OpenDXL

- Brokers abstracted away
- Event-based communication

Architecture of OpenDXL



- Brokers abstracted away
- Event-based communication

An instance of OpenDXL services

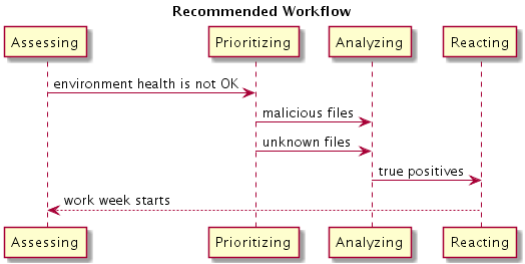
TIE's features

coordination of activities involving

- assessment of the security threats
  of configuration files, certificates, unsigned or unknown files, etc.
- prioritisation of analysis steps
  focusing on malicious or unknown files
- customisation of security queries
  based on reputation-based data such as product or company names
- reaction to suspicious indicators

Documenting TIE

Semi-formal diagrams



Verbal recommendations

a client "must have permission to send messages to the `/mcafee/service/tie/reputation/set` *topic"*

Sounds good...in theory

In practice

- "Stuff" developed @McAfee works fine
  - McAfee provides the service
  - and clients
- but it's a SOA: 3rd-party clients misbehave sometimes
- hence, defensive programming of TIE services

Sounds good...in theory

In practice

- "Stuff" developed @McAfee works fine
  - McAfee provides the service
  - and clients
- but it's a SOA: 3rd-party clients misbehave sometimes
- hence, defensive programming of TIE services

Caveat

3rd-party code may not be available for analysis
Hence, post-mortem analysis of execution logs to identify misbehaviour and communicate it to 3rd-parties

A methodology

Idea

Adding more precision:

1. draw the protocol (global choreography)
2. turn the "drawing" into a behavioural type
3. project to component specs (local types)
4. turn local specs into state machines

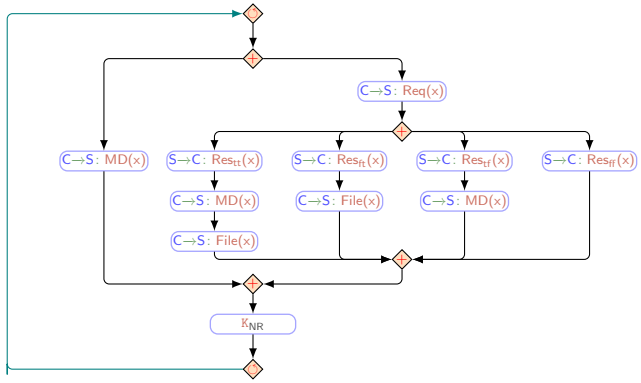
Advantages

- global choreographies: formal & precise (Pomset or Event Structure semantics[a]), yet intuitive
- algorithmically generate monitors
- enhance "program comprehension"

[a] See Ugo de'Liguoro's talk @ ICE 2020

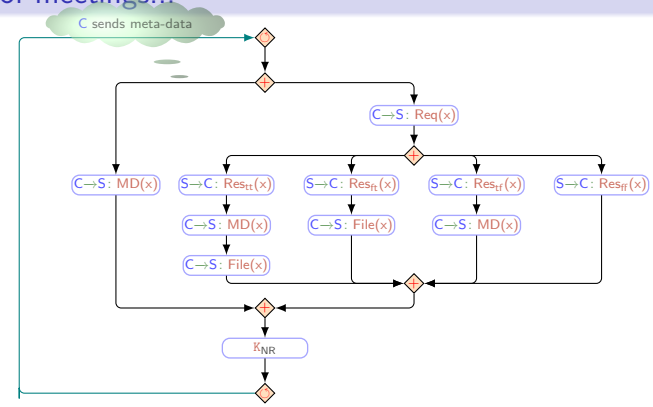
Let's tie our TIE

After a couple of meetings...



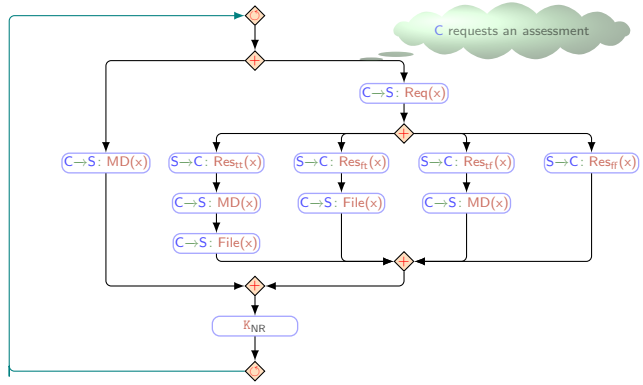
Let's tie our TIE

After a couple of meetings...



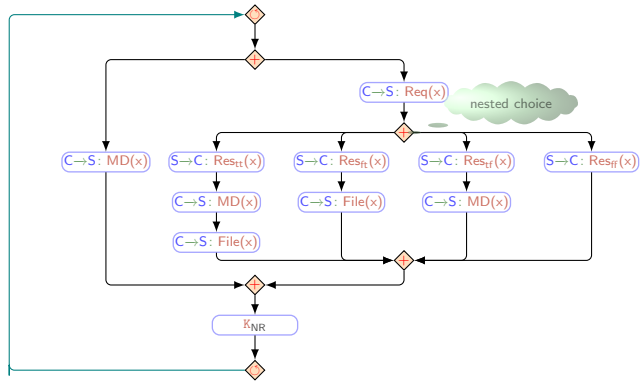
Let's tie our TIE

After a couple of meetings...



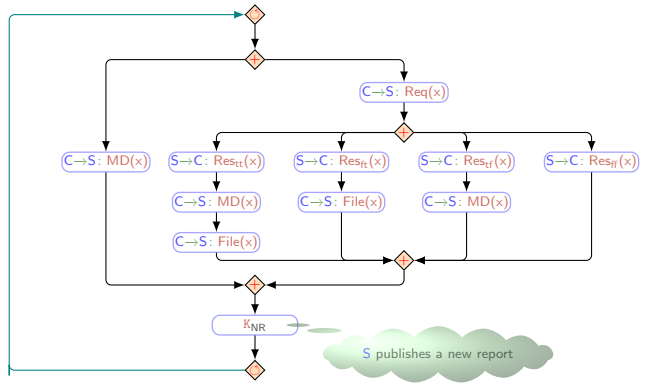
Let's tie our TIE

After a couple of meetings...



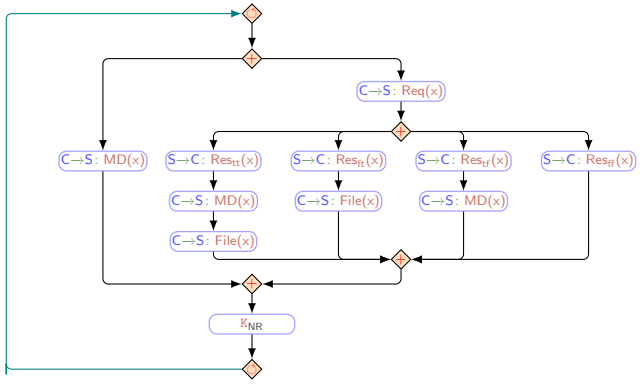
Let's tie our TIE

After a couple of meetings...



Let's tie our TIE

After a couple of meetings...



DISCLAIMER

No greek letters were used in the making of this global view

If you're versed in behavioural types

Behavioural types

Suitable devices for specification and analysis

- focus on control (mostly)
- assume point-to-point channels

If you're versed in behavioural types

Behavioural types

Suitable devices for specification and analysis

- focus on control (mostly)
- assume point-to-point channels

VS

Klaimographies

Behavioural types with

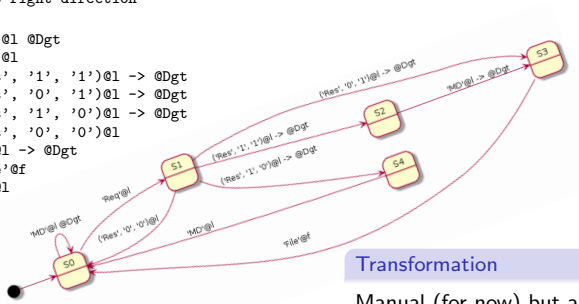
- focus on data (mostly)
- interactions based on generative communication
- unit & multi-roles

Monitors from projections

UML-like

```
@startuml left to right direction
[*] --> S0
S0 --> S0:  'MD' @l @Dgt
S0 --> S1:  'Req'@l
S1 --> S2:  ('Res', '1', '1')@l -> @Dgt
S1 --> S3:  ('Res', '0', '1')@l -> @Dgt
S1 --> S4:  ('Res', '1', '0')@l -> @Dgt
S1 --> S0:  ('Res', '0', '0')@l
S2 --> S3:  'MD'@l -> @Dgt
S3 --> S0:  'File'@f
S4 --> S0:  'MD'@l

@enduml
```



Transformation

Manual (for now) but algorithmic

- '@...' for data-flow analysis
- huge logs (can't fit memory)

Monitor checks expected data correspondences (e.g., 'file' corresponds to a 'file req')

Lessons learned

Effectiveness & Reproducibility (of (meta-)communication)

- non-deterministic & visual abstractions
  - help communication among stakeholders
  - provide insights & “inspirations”
- but semantics is necessary
  - to attain precision
  - to change mind
- a reviewer was “missing the difficulties in this formalisation”

Generality

- how tight to TIE are we?
- klaimographies were not designed *for* OpenDXL
- a reviewer noted: “event-based middleware are becoming the norm”
- choreography can go bottom-up (as noted by another reviewer)

Other FM?

Sure / but ...

- Model checking
  but it is not easy for lay-users to express properties in some temporal logic
- Other behavioural types
  usually too many greek letters
- Other FM (Petri nets, event structures,...)
  too low level (and (sadly) not much studied anymore)

What's next?

- Tool support (extend **ChorGram**)
  - validated global views ensure properties
  - automatise projections
  - code generation (eg TIE vs many versions/variants)
- Extend the application to other services of TIE / OpenDXL
- Klaimographies-inspired abstractions for CAS

Thank you
and
to the anonymous reviewers!