



Le génie pour l'industrie

AUTOMATIC GENERATION OF DESCRIPTION FILES FOR HIGHLY AVAILABLE SERVICES

Maxime Turenne

Ali Kanso

Abdelouahed Gherbi

6th International Workshop on Software
Engineering for Resilient Systems,
15th October 2014, Budapest

OUTLINE

- Introduction
 - What is High Availability (HA)
 - Current practice for achieving HA
- Background
 - The previous approach for generating middleware HA configuration
- A novel approach
 - New domain-specific modeling language
 - Our methodology for generating middleware HA configuration
- Prototype implementation
- Conclusion

HA DEFINITION

Service Availability (SA): the percentage of time the system/service is available throughout a period of time t.

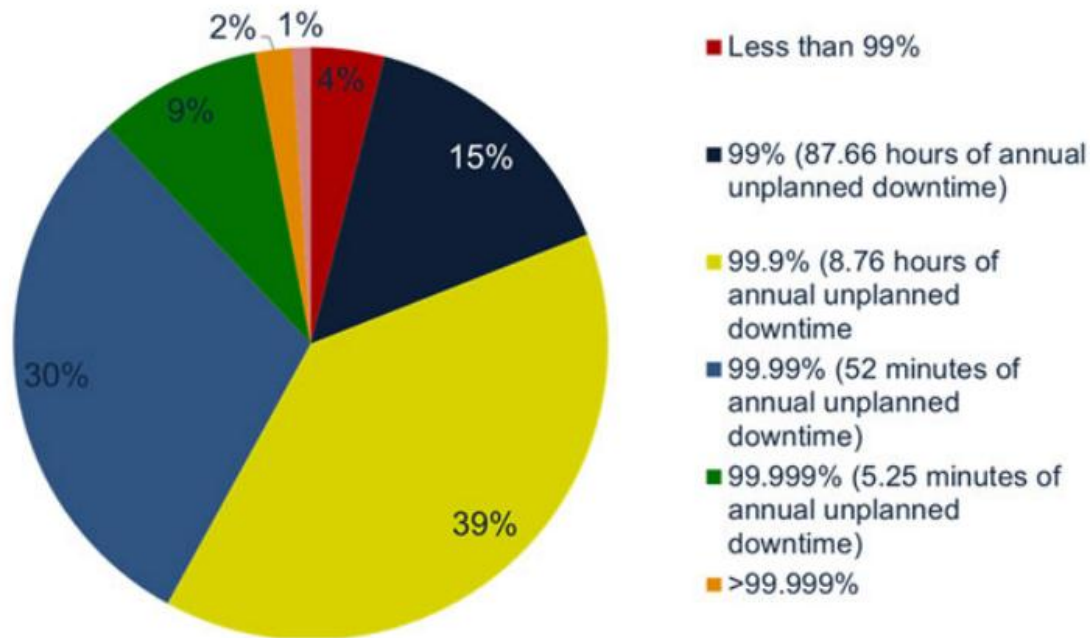
| Availability | Downtime per Year |
|--------------|-------------------|
| 90% | 36.5 days |
| 99% | 3.7 days |
| 99.9% | 8.8 hours |
| 99.99% | 52.6 min |
| 99.999% | 5.3 min |
| 99.9999% | 31.5 sec |



High Availability (HA): At least 99.999 % (a.k.a. five nines)

DEMAND ON HA

- More than **40%** of companies want **99.99%** availability ⇔ less than **one hour** outage per year



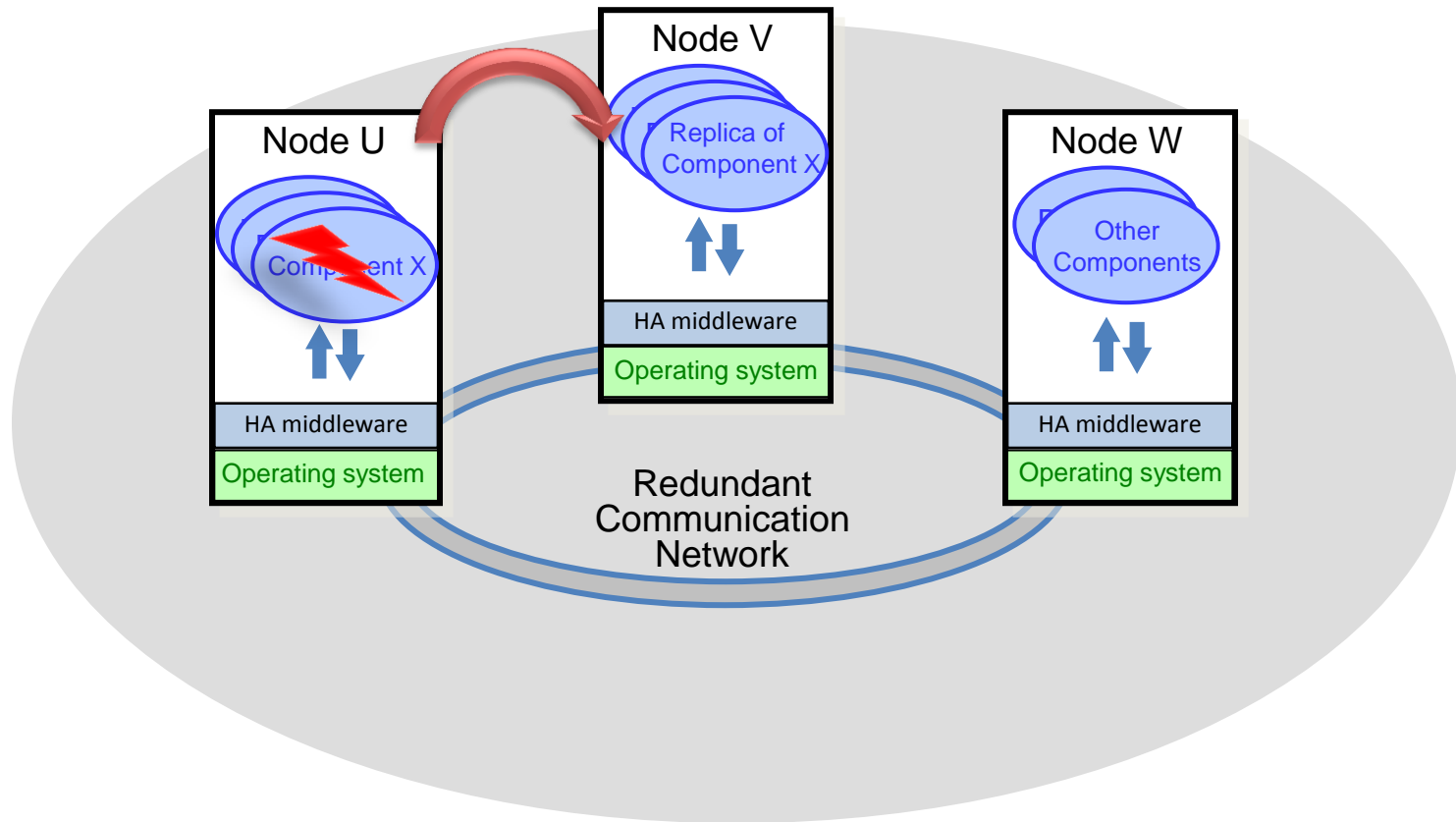
Information Technology and Intelligence Corp. survey

DOWNTIME COST

- 59% of Fortune 500 companies experience a minimum of 1.6 hours of downtime per week (Gartner 2011)
 - ⇔ 46,000,000\$ of loss per year (only employee's salary)
- A Ponemon Institute study shows that in the years 2012 and 2013, 91% of data centers endured unplanned outages
- Average loss of \$90,000 per hour in the media sector to about \$6.48 million per hour for large online brokerages

| | Total(Hour) | Average(Hour) | Availability | Cost/Hour(USD) | Cost(USD) |
|----------------|-------------|---------------|--------------|----------------|------------|
| 1. Amadeus | 1 | 0.167 | 99.998% | 89,000 | 89,000 |
| 4. Paypal | 5 | 0.833 | 99.990% | 225,000 | 1,125,000 |
| 5. Google | 5 | 0.833 | 99.990% | 200,000 | 1,000,000 |
| 6. Yahoo! | 6 | 1.000 | 99.989% | 200,000 | 1,200,000 |
| 7. Twitter | 7 | 1.167 | 99.987% | 200,000 | 1,400,000 |
| 8. Amazon | 24 | 4.000 | 99.954% | 180,000 | 4,320,000 |
| 9. Microsoft | 31 | 5.167 | 99.941% | 200,000 | 6,200,000 |
| 10. Hostway | 72 | 12.000 | 99.863% | 100,000 | 7,200,000 |
| 11. BlackBerry | 72 | 12.000 | 99.863% | 200,000 | 14,400,000 |
| 12. NaviSite | 168 | 28.000 | 99.680% | 100,000 | 16,800,000 |
| 13. OVH | 170 | 28.333 | 99.677% | 100,000 | 17,000,000 |
| Total | 568 | 94.667 | 99.917% | | 71,734,000 |

MIDDLEWARE BASED HA SOLUTIONS



*E.g.: **OpenSAF** is an open source implementation of an HA middleware (www.OpenSAF.org), with contributions from world leading telecom and computing companies.*

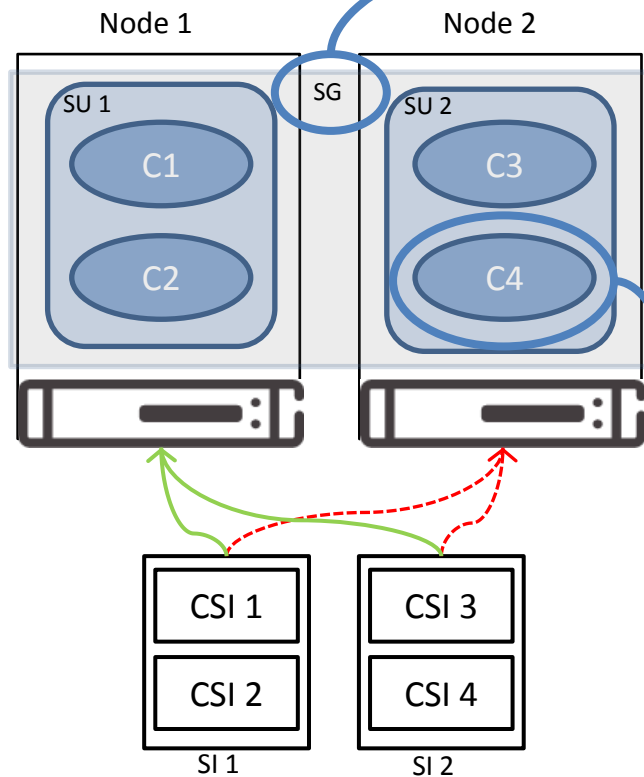
SAFORUM

(SERVICE AVAILABILITY FORUM)

- Consortium of industry-leading IT and Telecom companies.
- Defines open standards for HA systems:
 - Application Programming Interfaces
 - Guidelines for HA system
 - Specifications for an HA middleware

HA MW CONFIGURATION

- The HA management is performed based on a complex XML configuration file:



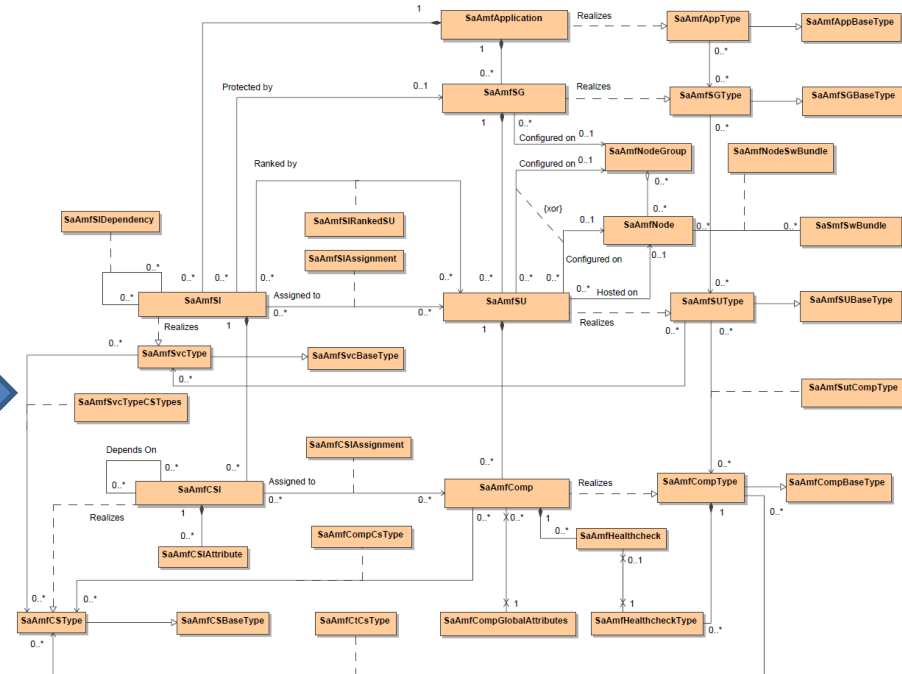
```
<object class="SaAmfSG">
  <attr>
    <name>saAmfSg=2N,safApp=net-java</dn>
  </attr>
  <attr>
    <name>saAmfSGType</name>
    <value>safVersion=1,safSgType=2N-net-java</value>
  </attr>
  <attr>
    <name>saAmfSGNumPrefActiveSUs</name>
    <value>1</value>
  </attr>
  <attr>
    <name>saAmfSGNumPrefStandbySUs</name>
    <value>1</value>
  </attr>
  <attr>
    <name>saAmfSGNumPrefInserviceSUs</name>
    <value>3</value>
  </attr>
  <attr>
    <name>saAmfSGNumPrefAssignedSUs</name>
    <value>3</value>
  </attr>
  <attr>
    <name>saAmfCtDefDisableRestart</name>
    <value>0</value>
  </attr>
  <attr>
    <name>saAmfCtDefCleanupCmdArgv</name>
    <value>stop</value>
  </attr>
  <attr>
    <name>saAmfCtDefClcCliTimeout</name>
    <value>10000000000</value>
  </attr>
  <attr>
    <name>saAmfCtCompCategory</name>
    <value>8</value>
  </attr>
</object>
```


HA MW CONFIGURATION

- The configuration structure is described using a standardized UML class diagram:

```
imm.xml_workingbackup (/etc/opensaf) - gedit
Open Save Undo Redo
imm.xml_workingbackup
13347 <name>smfClusterRebootCmd</name>
13348 <value>/usr/local/lib/opensaf/smf-cluster-reboot</value>
13349 </attr>
13350 <attr>
13351 <name>smfCliTimeout</name>
13352 <value>600000000000</value>
13353 </attr>
13354 <attr>
13355 <name>smfCampMaxRestart</name>
13356 <value>10</value>
13357 </attr>
13358 <attr>
13359 <name>smfBundleCheckCmd</name>
13360 <value>/usr/local/lib/opensaf/smf-bundle-check</value>
13361 </attr>
13362 <attr>
13363 <name>smfBackupCreateCmd</name>
13364 <value>/usr/local/lib/opensaf/smf-backup-create</value>
13365 </attr>
13366 <attr>
13367 <name>smfAdminOpTimeout</name>
13368 <value>600000000000</value>
13369 </attr>
13370 <attr>
13371 <name>SaImmAttrImplementerName</name>
13372 <value>safSmfService</value>
13373 </attr>
13374 <attr>
13375 <name>SaImmAttrClassName</name>
13376 <value>OpenSafSmfConfig</value>
13377 </attr>
13378 <attr>
13379 <name>SaImmAttrAdminOwnerName</name>
13380 <value>IMMLOADER</value>
13381 </attr>
13382 </object>
13383 </imm:IMM-contents>
```

Based on



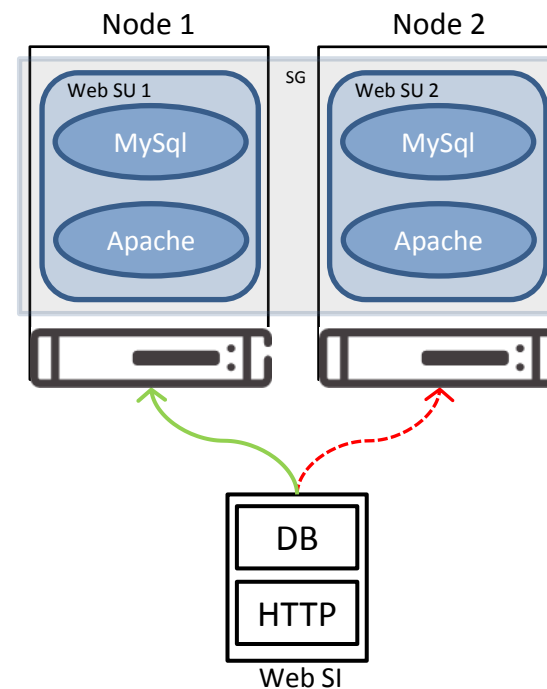
Configuration model

COMPLEX DOMAIN DETAILS

- The 2 main category in the Availability Management Framework (AMF) configuration:

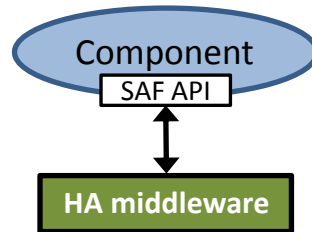
– The Service Provider:

– The Service:

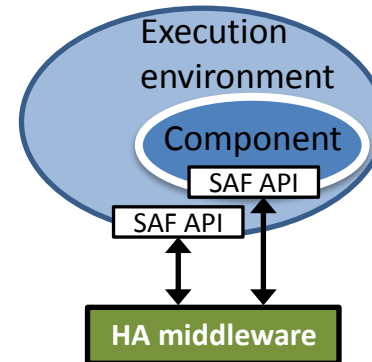


SERVICE PROVIDER

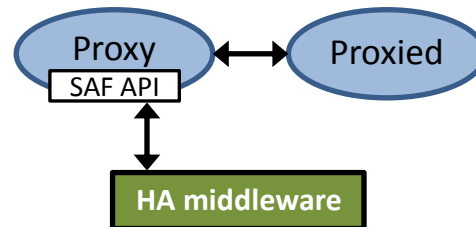
- SaAware:



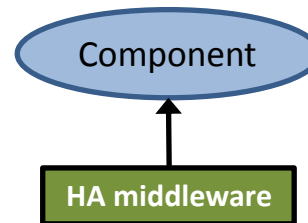
- Container/Contained:



- Proxy/Proxied:

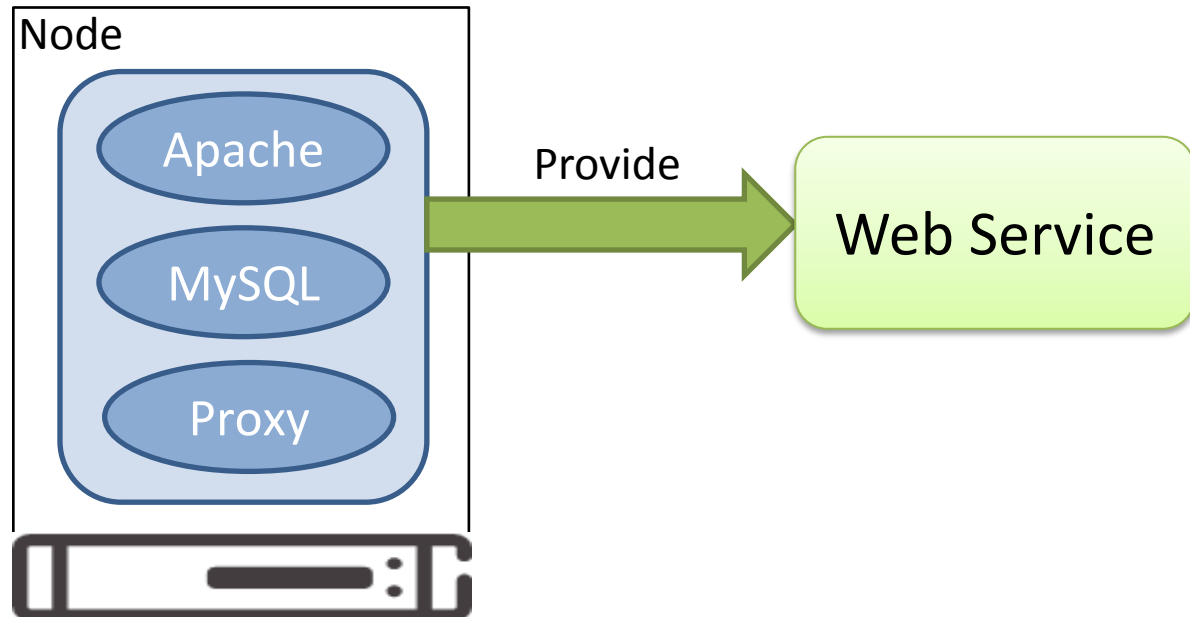


- NonSaAwareNonProxied:



HIERARCHICAL COMPOSITION

- This HA MW support the notion of multiple inter-dependent components collaborating to provide a higher level of service:



OUTLINE

✓ Introduction

- What is High Availability (HA)
- Current practice for achieving HA

• Background



- The previous approach for generating middleware HA configuration

• A novel approach

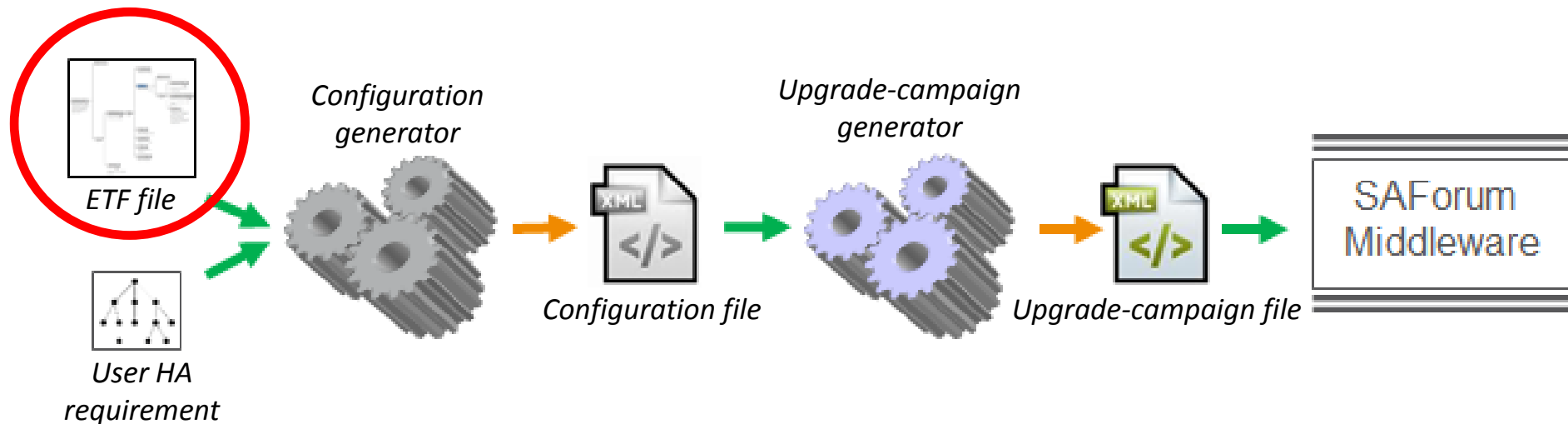
- New domain-specific modeling language
- Our methodology for generating middleware HA configuration

• Prototype implementation

• Conclusion

PREVIOUS APPROACH

- Previous automatic configuration approach:

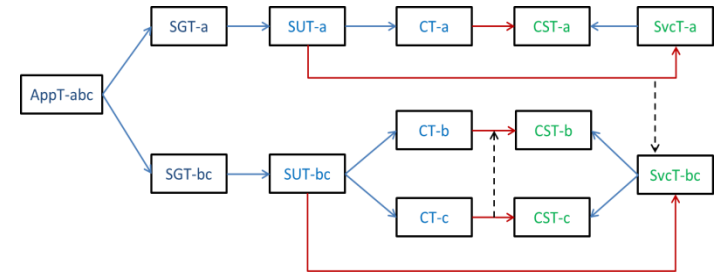


*A. Kanso, A. Hamou-Lhadj, M. Toeroe, and F. Khendek, "Generating AMF Configurations from Software Vendor Constraints and User Requirements", in Proc. of the Forth International Conference on Availability, Reliability and Security, Fukuoka, Japan, 2009, pp. 454-461

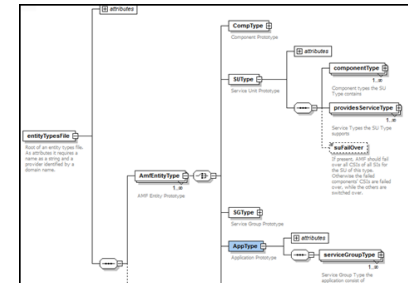
ENTITY TYPE FILE (ETF)

- Software vendor description for:

- Software capabilities
- Dependencies
- Limitations



- Standardized by an XML schema



ETF schema

- With constraints derived from:

- the XML schema,
- the Software Management Framework specification,
- the Availability Management Framework specification

```
context ContainedComponent
invariant C_CT1: self.componentServiceProvides
->forAll(cs : ComponentService | cs.needs
->forAll(cs2 : ComponentService | cs2.ProvidedBy
->forAll(c : Component | c.ocIsTypeOf(ContainedComponent))));

context ContainedComponent
invariant C_CT2: self.componentServiceProvides
->forAll(cs : ComponentService | cs.neededBy
->forAll(cs2 : ComponentService | cs2.ProvidedBy
->forAll(c : Component | c.ocIsTypeOf(ContainedComponent))));
```


CHALLENGES OF DEFINING AN ETF FILE

- The user needs to write the XML file manually,
- Domain constraints are informally described in thousand of specification pages,
- Therefore, the user needs deep domain knowledge.

OUTLINE

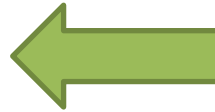
✓ Introduction

- What is High Availability (HA)
- Current practice for achieving HA

✓ Background

- The previous approach for generating middleware HA configuration

• A novel approach



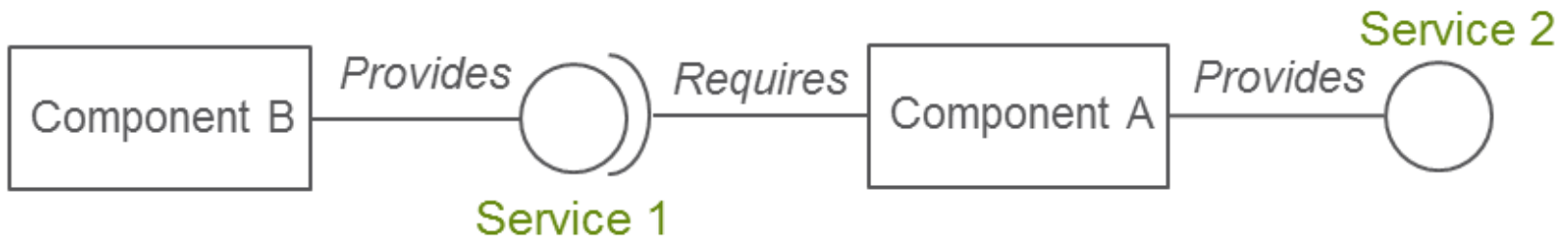
- New domain-specific modeling language
- Our methodology for generating middleware HA configuration

• Prototype implementation

• Conclusion

ABSTRACTING THE DOMAIN

- We designed a high level modeling language that is:
 - Graphical
 - Intuitive
 - Expressive
 - Standards-based
- We decided to extend the UML component diagram:



NEW CONSTRUCTS

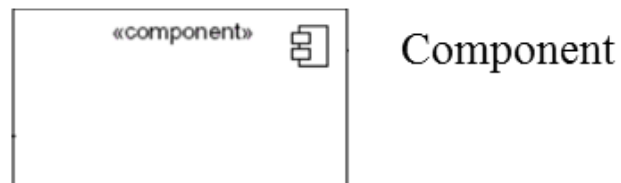
UML component diagram

main constructs:

-----> Depends on

○ Provides Interface

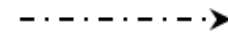
—) Requires Interface




Added Constructs:


Added Dependency:

Interface Colocation Dependency:




Added Interfaces:

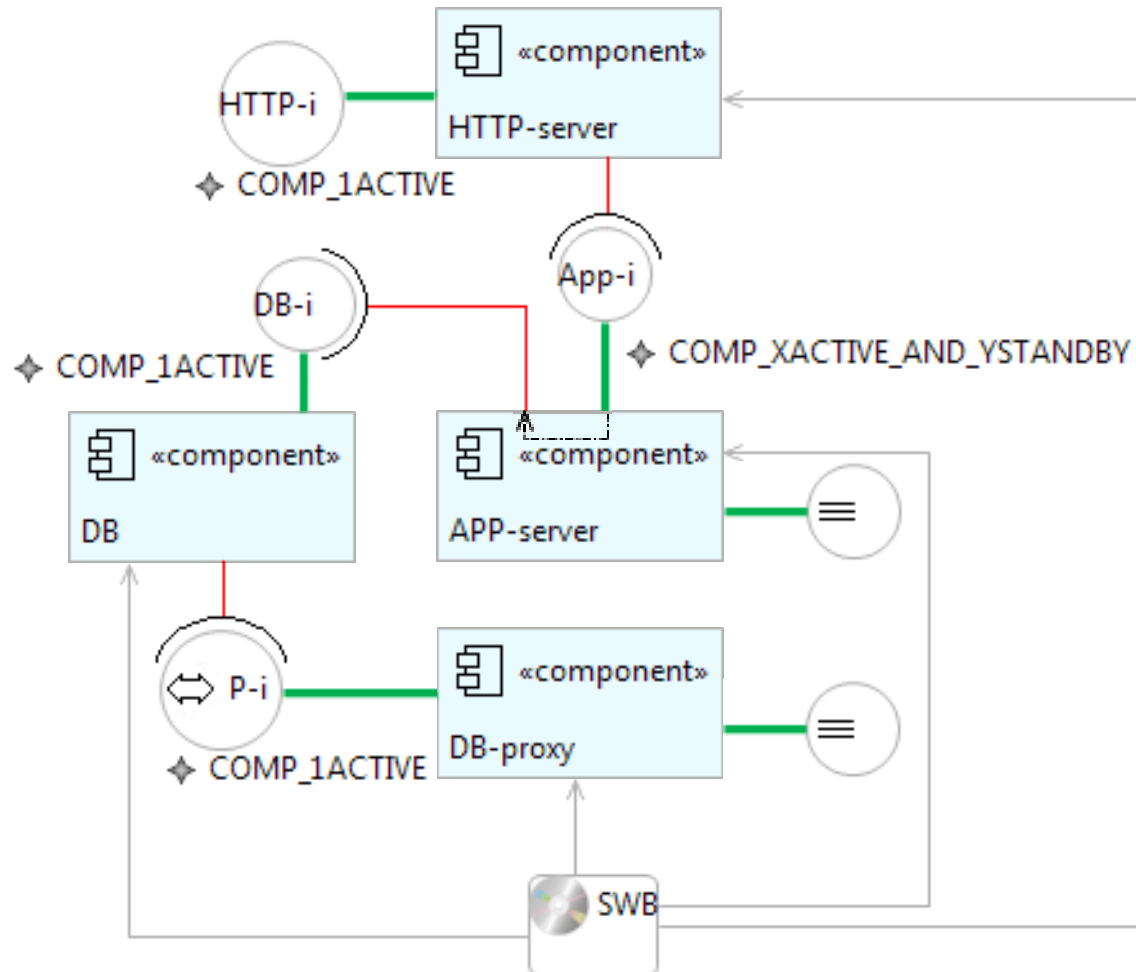
Proxy Interface:  —

Container Interface:  —

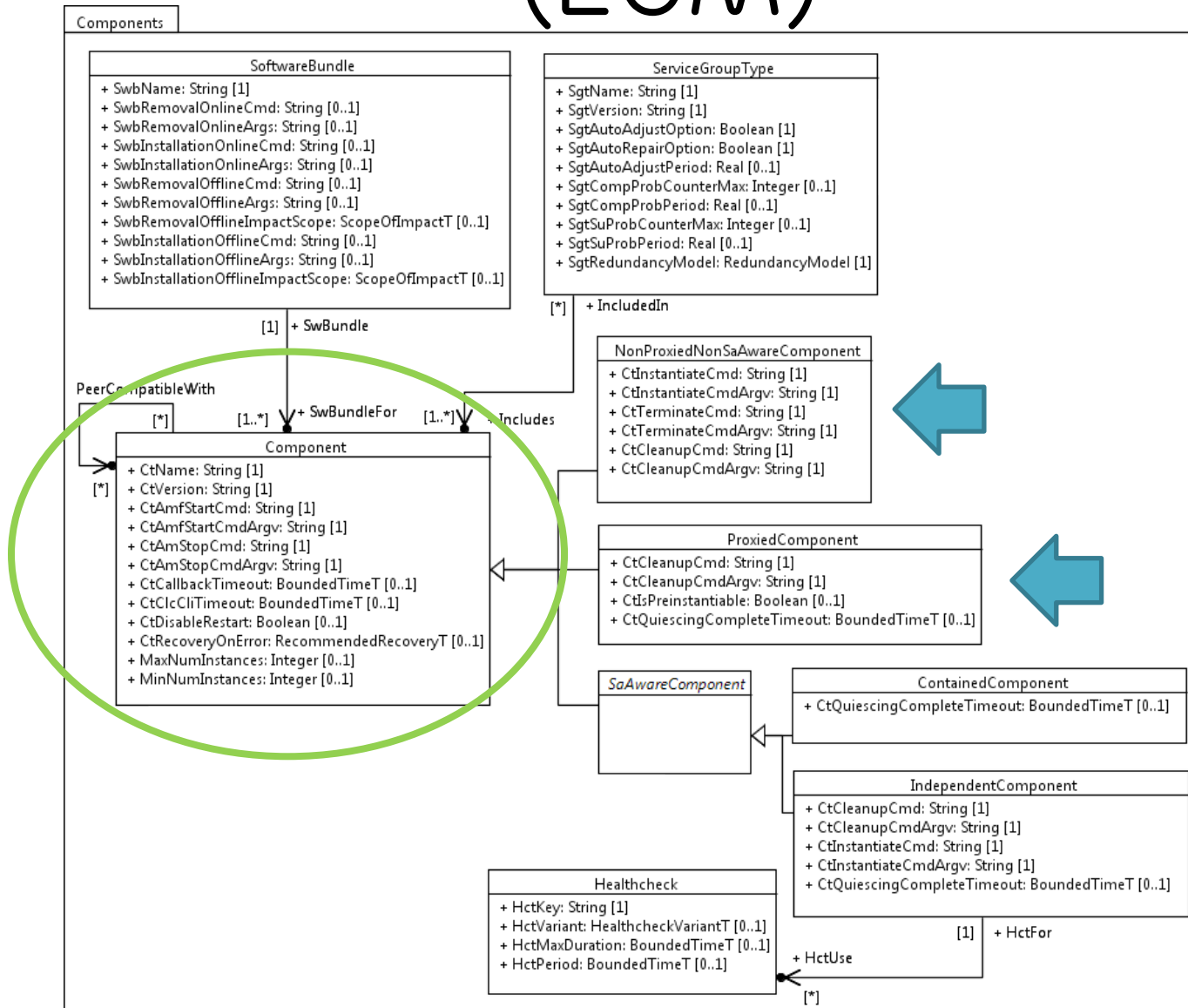
SAF Interface:  —

Checkpoint Interface:  —

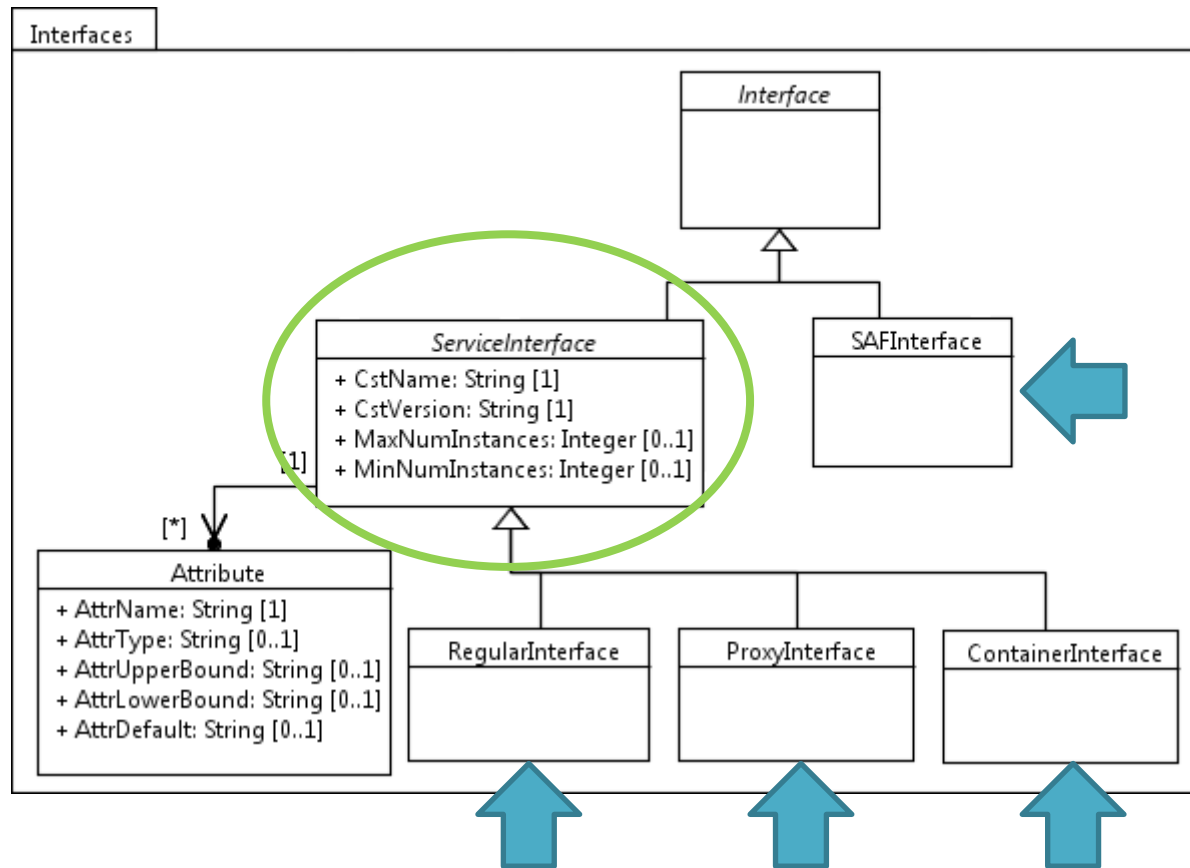
NEW MODELING LANGUAGE



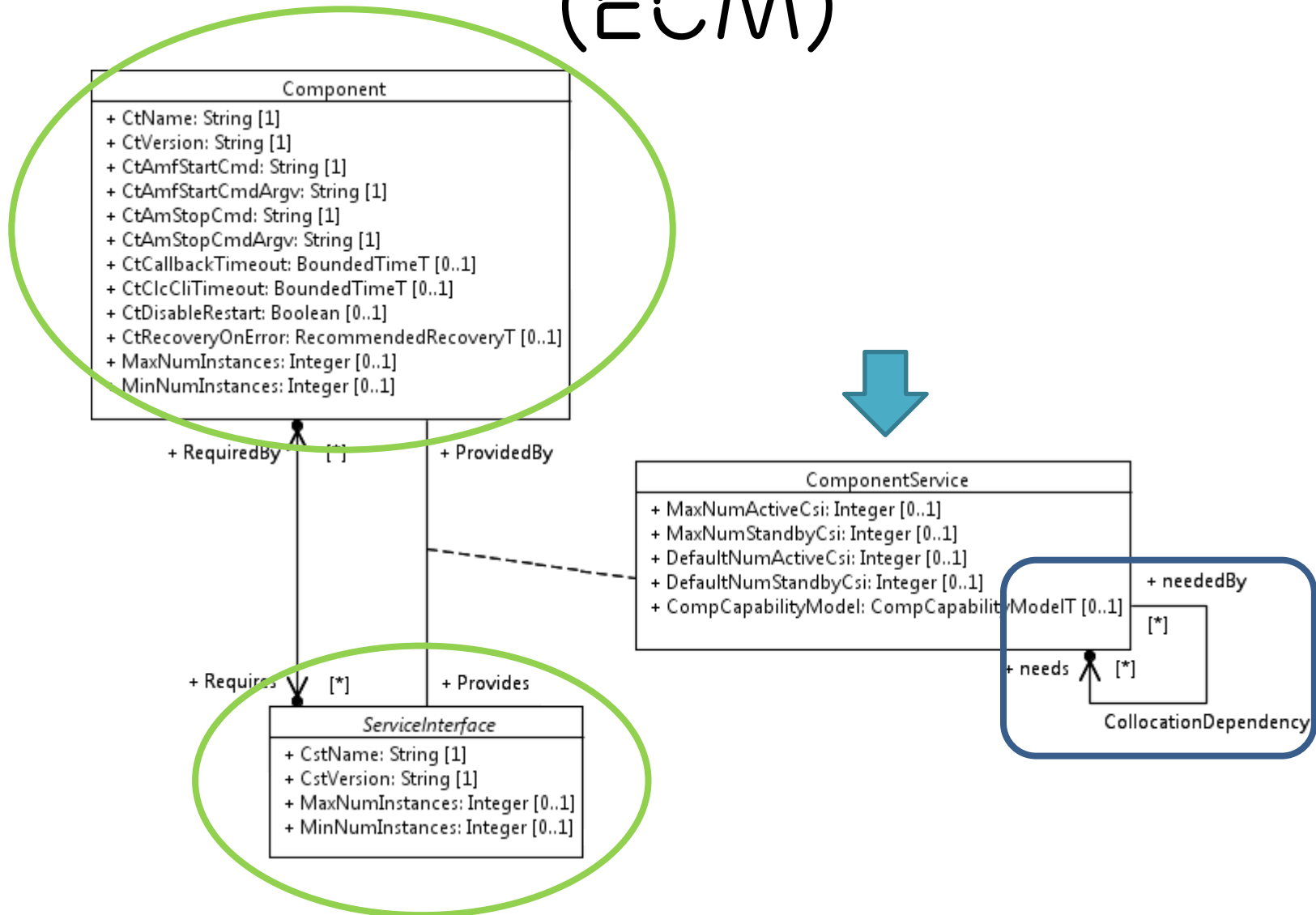
ETF COMPONENT DIAGRAM (ECM)



ETF COMPONENT DIAGRAM (ECM)



ETF COMPONENT DIAGRAM (ECM)

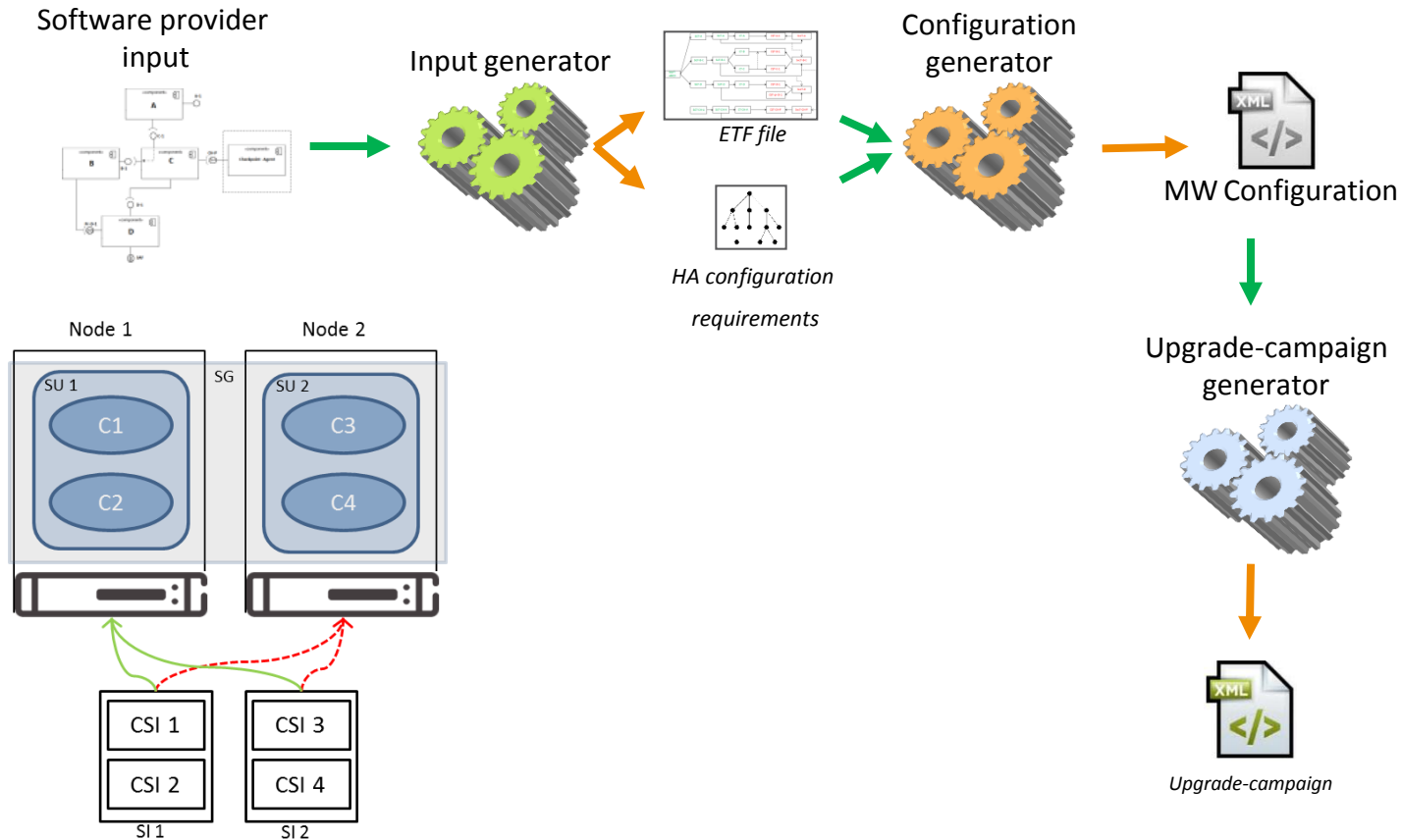


DOMAIN CONSTRAINTS

- Our model is annotated with dozens of OCL constraints.
- One of the OCL constraints:

```
invariant P_CT1: self.componentServiceProvides
->select(cs : ComponentService | cs.CompCapabilityModel <> CompCapabilityModelT::COMP_NON_PRE_INSTANTIABLE)
->size() > 0 implies self.componentServiceProvides
->forAll(cs : ComponentService | cs.needs
  ->forAll(cs2 : ComponentService | cs2.Provides
    ->select(si : ServiceInterface | si.ocIsTypeOf(ProxyInterface))
    ->size() = 0)) and self.Requires
->select(si : ServiceInterface | si.ocIsTypeOf(ProxyInterface))
->forAll(pi : ServiceInterface | pi.componentServiceProvidedBy
  ->forAll(cs3 : ComponentService | cs3.ProvidedBy
    ->forAll(c : Component | c.componentServiceProvides
      ->forAll(cs4 : ComponentService | cs4.neededBy
        ->excludesAll(self.componentServiceProvides)))) and self.Requires
->select(si : ServiceInterface | si.ocIsTypeOf(ProxyInterface))
->forAll(pi : ServiceInterface | pi.componentServiceProvidedBy
  ->forAll(cs3 : ComponentService | cs3.ProvidedBy
    ->forAll(c : Component | c.componentServiceProvides
      ->forAll(cs4 : ComponentService | cs4.needs
        ->excludesAll(self.componentServiceProvides)))));
```

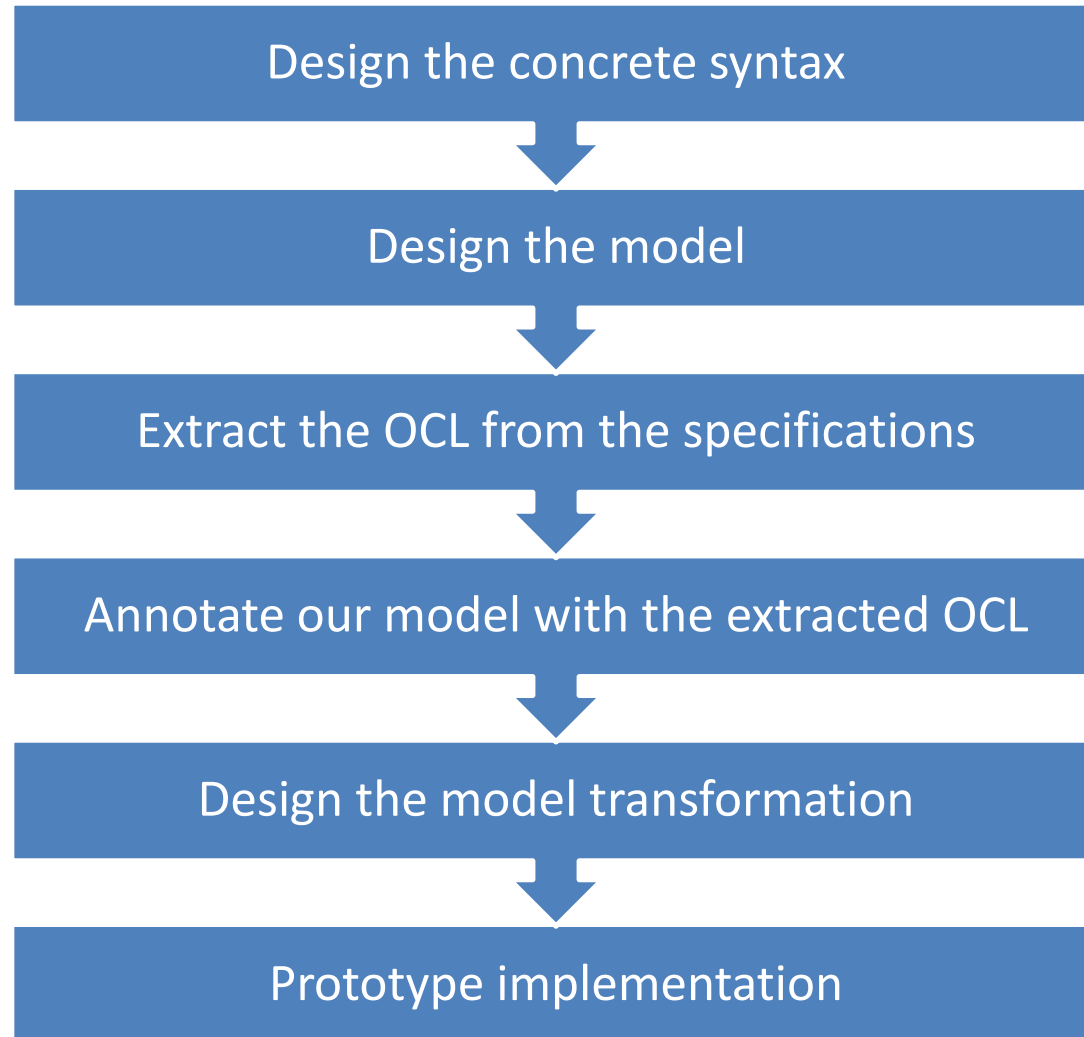

NOVEL APPROACH



openSAF Middleware

Software Management
Framework

RESEARCH PROCESS



OUTLINE

✓ Introduction

- What is High Availability (HA)
- Current practice for achieving HA

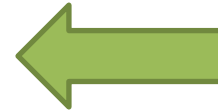
✓ Background

- The previous approach for generating middleware HA configuration

✓ A novel approach

- New domain-specific modeling language
- Our methodology for generating middleware HA configuration

• Prototype implementation



• Conclusion

WORKFLOW

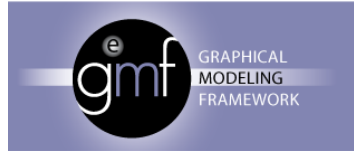
Design HA configuration

Trigger the automatic validation

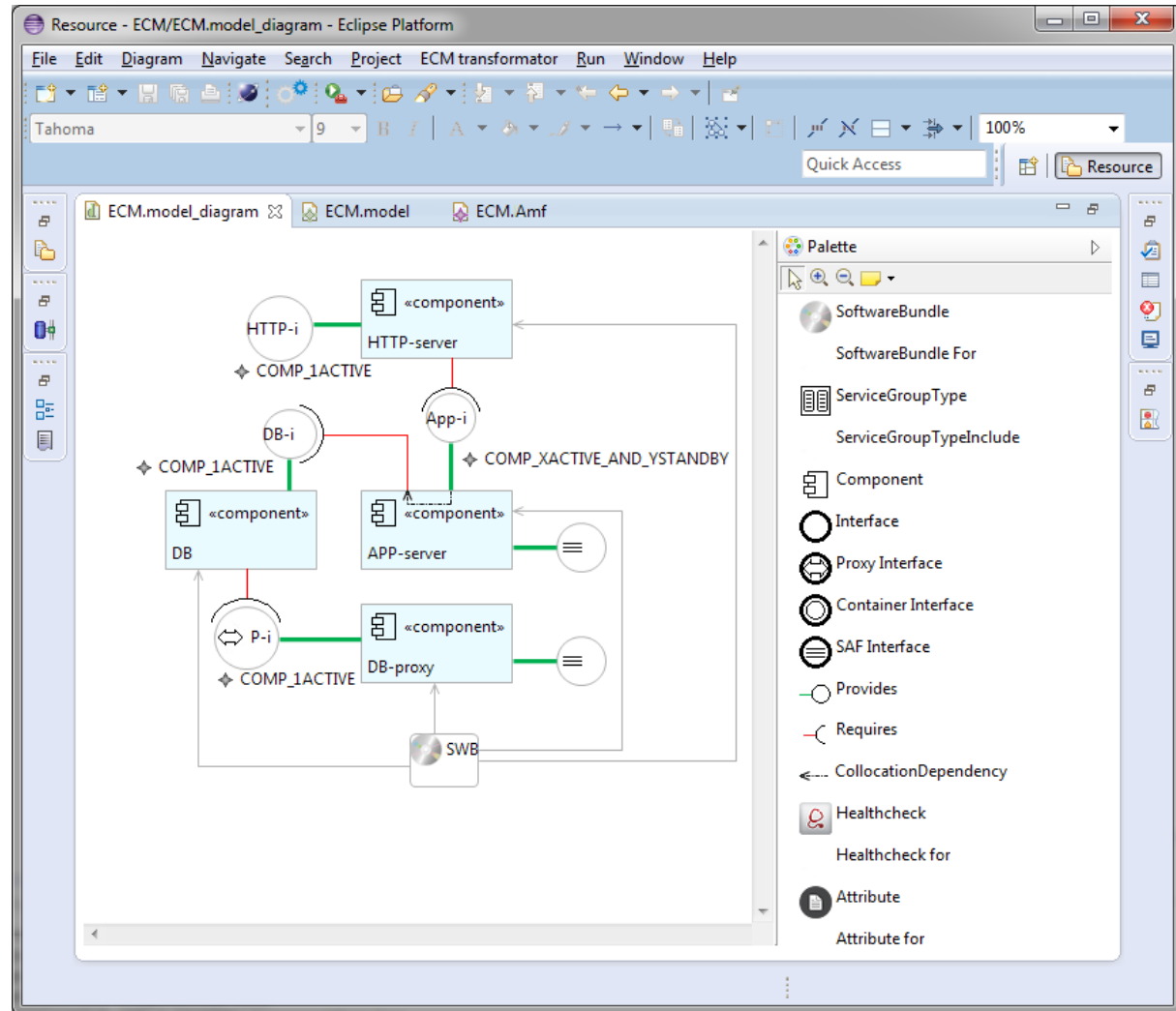
Trigger the automatic generation

Integrate the configuration into the
middleware

PROTOTYPE IMPLEMENTATION



OCLinEcore



CONCLUSION

- ✓ Reduce the design complexity of configurations
 - Using an intuitive language that saves time and effort.



- ✓ Reduce the configuration errors
 - By automatically validating the generated configurations against domain constraints.



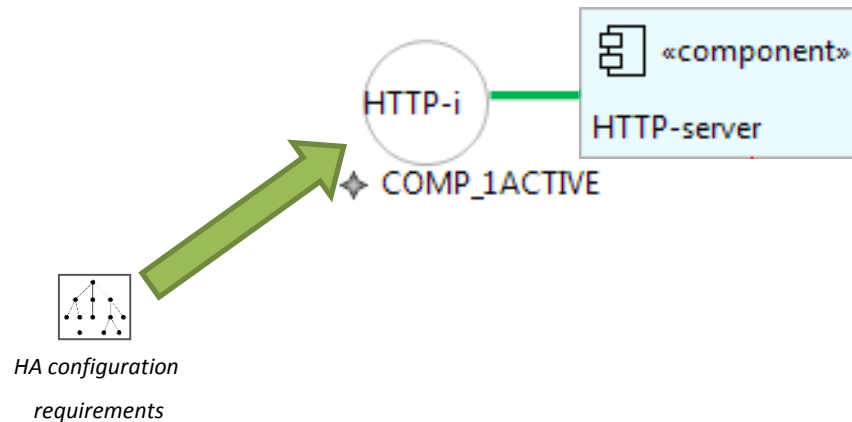
- ✓ No need for the developer to manually manipulate heavy and complex XML files.



- ✓ Abstraction of the domain complexity.

FUTURE WORK

- Integrate the specification of HA and non-functional requirements in our model and design language.



THANK YOU 😊