eXtensible Access Control Markup Language

Autor

Francisco Alexandre de Gouveia

Orientador

Doutor Diogo Gomes

Colaborador

Engenheiro Ricardo Azevedo





Universidade de Aveiro Instituto de Telecomunicações Portugal Telecom Inovação

XACML – O que é

- Norma definida pela OASIS para controlo de acessos extensível e genérico
- Consiste em:
 - Linguagem extensível de políticas em xml
 - Linguagem extensível de pergunta-resposta em xml
 - Arquitectura distribuída baseada em:
 - Policy Enforcement Point
 - Policy Decision Point
 - Policy Information Point
 - Policy Administration Point





XACML – Para que serve

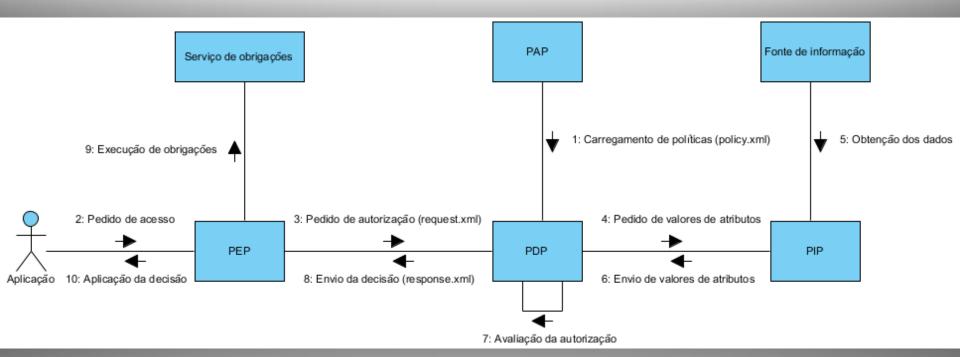
- Controlo de acessos
- Como é genérico e extensível, pode ser aplicado em qualquer contexto:
 - Controlo de acesso de portas
 - Controlo de acesso numa página web
 - Controlo de acesso de serviços
- Só toma decisões!
 - Não serve para saber que permissões uma determinada entidade tem





XACML - Como funciona

- Sistema de pergunta resposta
- Arquitectura distribuída





XACML - Pergunta

- O sujeito pode realizar a acção no recurso num determinado contexto?
- Palavras chave:
 - Sujeito Tanto pode ser uma pessoa como um equipamento
 - Acção Predicado (aceder, ler, editar, abrir,...)
 - Recurso O destino da acção (serviço, publicação, texto, porta, ...)
 - Contexto Tudo o que não se enquadre nos anteriores (espaço temporal, níveis de carga de processamento, nº de pessoas num local)





XACML - Pergunta

 O sujeito pode realizar a acção no recurso num determinado contexto?

```
<Request>
                                                                                                                                                                                                                                                                                                                         Exemplo em XACMLv2
                      <Subject>
                                            <a href="#"><Attribute AttributeId="utilizador"</a>
                                            DataType="http://www.w3.org/2001/XMLSchema#string">
                                                                                                                                                                                                                                                                                                                          Sujeito
                                                                   <AttributeValue>Anónimo</AttributeValue>
                                            </Attribute>
                      </Subject>
                      <Action>
                                            <a href="mailto:</a></a></a><a href="https://www.action.id"</a><a href="mailto:action:action.id"</a><a href="mailto:action:action.id"</a><a href="mailto:action:action.id"</a><a href="mailto:action:action.id"</a><a href="mailto:action:action.id"<a href="mailto:action:action.id"<a href="mailto:action:action.id"<a href="mailto:action:action.id"<a href="mailto:action:action.id"<a href="mailto:action:action.id"<a href="mailto:action.id"</a><a href="mailto:action.id"<a href="mailto:action.id">action.id<a href="mailto:action.id"<a href="mailto:action.id">action.id<a href="mailto:action.id">action.id<a href="mailto:action.id"<a href="mailto:action.id">action.id<a href="mailto:act
                                            DataType="http://www.w3.org/2001/XMLSchema#string">
                                                                                                                                                                                                                                                                                                                                                                    Acção
                                                                   <a href="#"><AttributeValue>Ler</a></a>
                                            </Attribute>
                      </Action>
                      <Resource>
                                            <a href="http://www.w3.org/2001/XMLSchema#string">
                                                                   <a href="#"><AttributeValue>Tópico</a>/AttributeValue>
                                            </Attribute>
                      </Resource>
                                                                                                                                                                                                                                                                                                                           Recurso
</Request>
```

XACML - Pergunta

 O sujeito pode realizar a acção no recurso num determinado contexto?



XACML - Resposta

- Tipos de resposta:
 - Permissão concedida (Permit)
 - Permissão negada (Deny)
 - Permissão indeterminada (Indeterminate)
 - Não existem políticas aplicáveis (Not applicable)
- Juntamente com:
 - Tarefas a serem executadas antes de ser permitido o acesso *





XACML - Resposta

- * Na versão 2 do XACML, existe o elemento Obligation
- O Policy Enforcement Point deve executar todas as tarefas descritas nos Obligations

• O que acontece quando a tarefa não é relevante e o Policy Enforcement Point não a consegue executar?

XACML - Resposta

- * Na versão 3 do XACML, existem os elementos Obligation e Advice
- O Policy Enforcement Point :
 - Deve executar todas as tarefas descritas nos Obligations
 - Tentar executar as tarefas descritas nos Advices

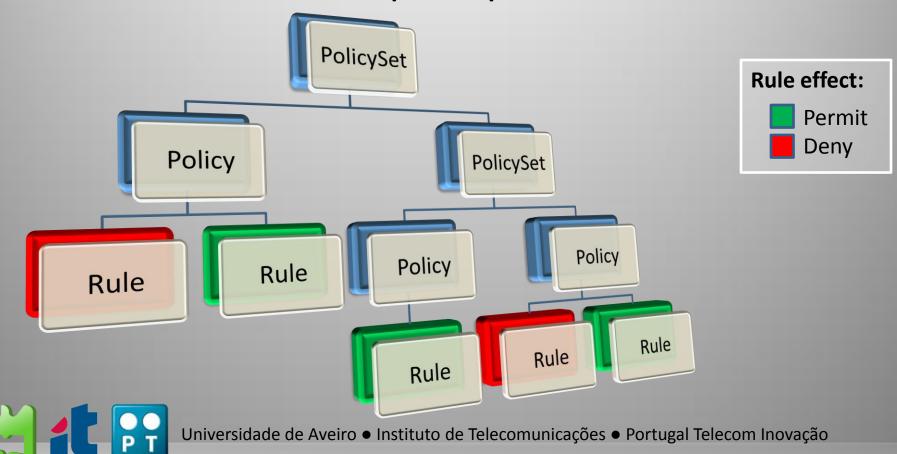
 Se uma tarefa do elemento Advice falhar, a decisão não é alterada



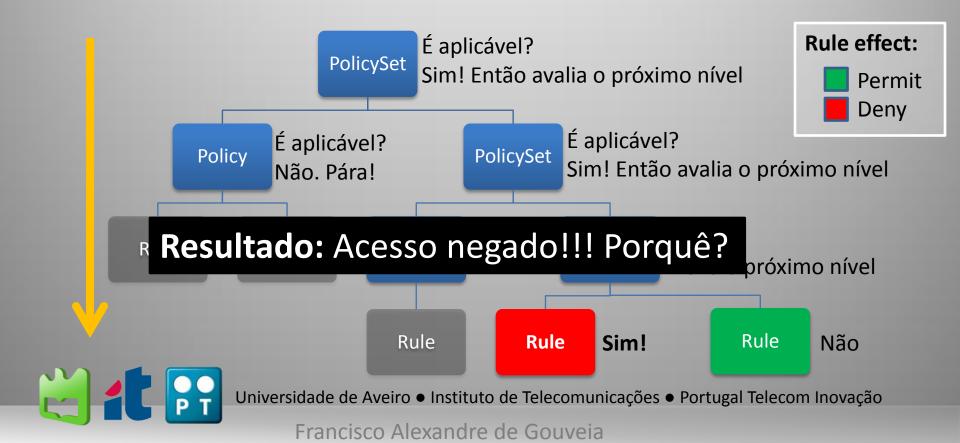




- Como são estruturadas as políticas?
- Existem 3 elementos principais



- Como são avaliados os elementos?
- Cada elemento tem um "Target"
- Avaliação feita a partir do topo

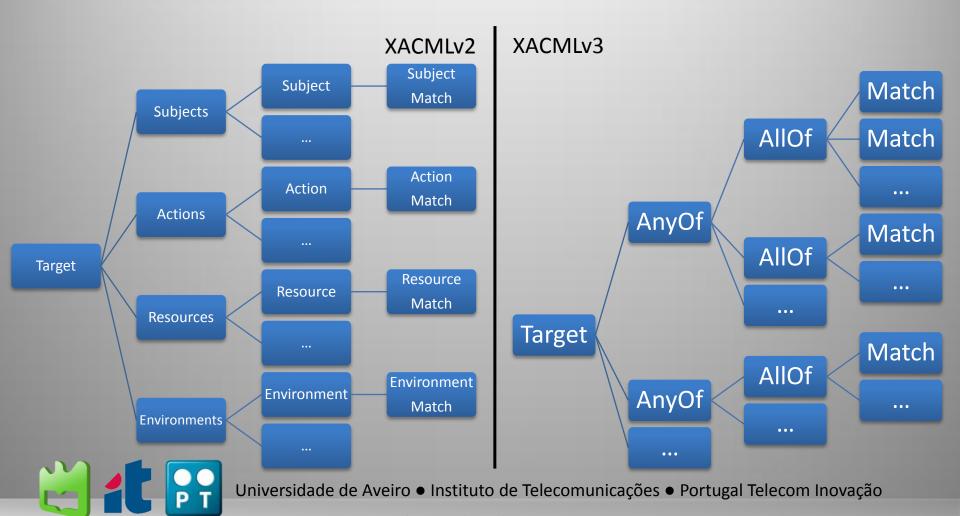


- E se mais do que uma regra/política for aplicável?
- Resposta: Algoritmos combinatórios
 - Permit-overrides
 - Deny-overrides
 - Only-one-applicable
 - First-applicable
 - ... (podem ser criados outros)





Como é constituído um Target?



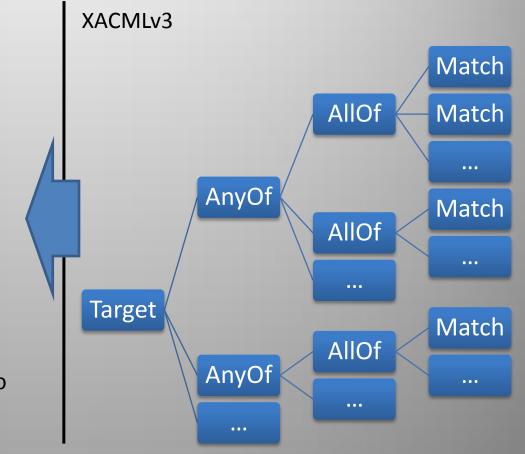
Francisco Alexandre de Gouveia

Como é constituído um Target?

- Processo uniforme para todas as categorias
- •Permite uniões e intersecções

Mas...

- •Elementos AnyOf e AllOf não têm atributos identificadores
 - •Na gestão de políticas implica analisar os valores contidos para saber qual editar/apagar e onde inserir
 - •Ou recriar Target por cada alteração







PAP XACMLv3 Policy Administration Point

Autor

Francisco Alexandre de Gouveia

Orientador

Doutor Diogo Gomes

Colaborador

Engenheiro Ricardo Azevedo





Universidade de Aveiro Instituto de Telecomunicações Portugal Telecom Inovação

- Objectivos deste projecto
 - Sistema de informação extensível (importação de módulos sem recompilação)
 - Interface de administrador que abstraia a complexidade do XACMLv3
 - Criação de políticas que obedeçam à norma

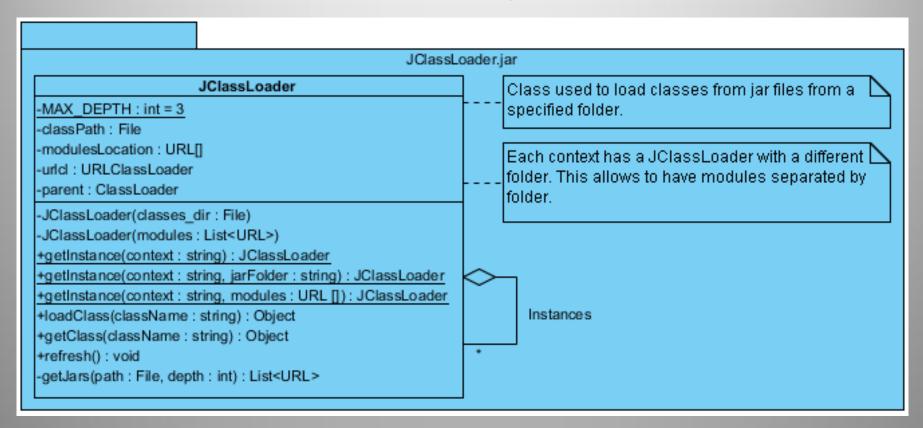




- Extensibilidade
 - Permitir importação de classes que implementem as interfaces definidas para:
 - Obtenção de informação do Policy Information Point
 - Persistência e obtenção de políticas
- Solução usada
 - URLClassLoader, carrega classes de ficheiros *.jar,
 em tempo de execução



• Extensibilidade (Classe de carregamento de classes)





Extensibilidade (Interfaces)

<<Interface>> **IPolicyRetreiver** +getRootPolicy(depth : int) : Node +getPolicyTreeElement(id : string, depth : int) : Node +getPolicySet(policySetId : string, depth : int) : Node +getPolicy(policyId: string, depth:int): Node +getRule(ruleId: string, int depth): Node +insertElementIntoPolicySetAsFirst(policySetId: string, element: Node): OperationResult +insertElementIntoPolicySetAsLast(policySetId: string, element: Node): OperationResult +insertElementIntoPolicySetAfterElement(policySetId: string, elementId: string, element: node): OperationResult +insertElementIntoPolicyAsFirst(policyId: string, element: Node): OperationResult +insertElementIntoPolicyAsLast(policyId: string): OperationResult +insertElementIntoPolicyAfterElement(policyId: string, elementId: string, element: Node): OperationResult +removeElementFromPolicyTreeElement(elementId: string, elementName: string): OperationResult +removePolicySet(policySetId: string): OperationResult +removePolicy(policyld: string): OperationResult +removeRule(ruleld : string) : OperationResult +policySetExist(policySetId: string): boolean +policyExist(policyld : string) : boolean +ruleExist(ruleId: string): boolean

<<Interface>> IInfoRetreiver

+getResourceDescription(id : string) : string +getResourceShortName(id : string) : string

+listResources(): Set<string>

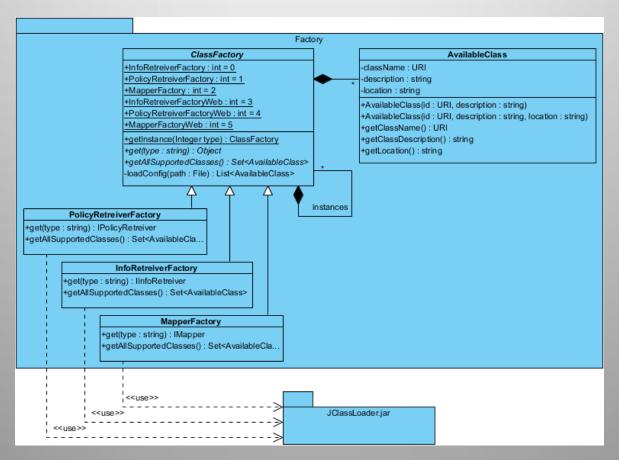
+listResources(category : string) : Set<string>

+doesMapping(): boolean

+setMapper(mapping : IMapper) : OperationResult



Extensibilidade (Factory)

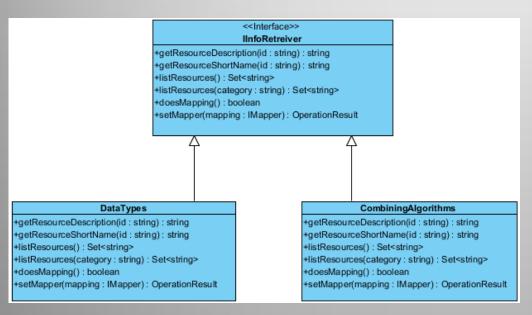






Interfaces <<interface>> Polic IPolicyRetreiver getRoat Policy(depth: int): Node getPolicyTreeElement(id: string, depth: int): Node getPolicySet(policySetId : string, depth : int) : Node w3.org.dom +getPolicy(policyld:string,depth:int): Node getRule(ruleId : string, int depth) : Node «Interface» «Interface» insertElementIntoPolicySetAsFirst(policySetId: string, element: Node): OperationResult NamedNodeMap insertElement IntoPolicySetAst.ast/policySetId: string, element: Node): OperationResult +insertElementIntoPolicySetAfterElement(policySetId:string, elementId:string, element:node): OperationResult +insertElement IntoPolicyAsFirst(policyId: string, element: Node): OperationResult +insertElementIntoPolicyAsLast(policyId:string): OperationResult Exte +insertElement IntoPolicyAfterElement(policyId: string, elementId: string, element: Node): OperationResult removeElementFromPolicyTreeElement(elementId: string, elementName: string): OperationResult removePolicySet(policySetId:string): OperationResult +removePolicy(policyld: string): OperationResult +removeRule(ruleId : string) : OperationResult. +policySetExist(policySetId : string) : boolean +policyExist(policyld : string) : boolean +ruleExist(ruleId : string) : boolean gerPolicySellad starting, despits in this Note gerPolicySellad starting, despits in the Note gerPolicySellad starting, despits in 19. Note gerPolicySellad starting, set despits 19. Note gerPolicySellad starting, set despits 19. Note gerPolicySellad starting, set despits 19. Note in set Elementhis PolicySellad starting policySellad starting, set in set Elementhis PolicySellad starting policySellad starting, set in set Elementhis PolicySellad starting of Sellad starting in set in set Elementhis PolicySellad Sellad starting in Sellad starting, set in set Elementhis PolicySellad Elementhis Sellad starting, in Sellad starting, set in set in set Elementhis PolicySellad Elementhis Sellad starting, in Sel PolicyRetreiver XACMLNamedNodeMap PolicyRetreiver +getRootPolicy(depth:int):Node +getPolicyTreeElement(id: string, depth: int): Node +getPolicySet(policySetId : string, depth : int) : Node +getPolicy(palicyld:string,depth:int):Node +getRule(ruleId : string, int depth) : Node +insertElement IntoPolicySetAsFirst(policySetId: string, element: Node): OperationResult 3 DB XML +insertElement IntoPolicySetAst.ast(policySetId: string, element: Node): OperationResult. +insertElementIntoPalicySetAfterElement(palicySetId: string, elementId: string, element: node): OperationResult +insertElementIntoPalicyAsFirst(palicyld: string, element: Nade): OperationResult ««use» +insertElementIntoPolicyAsLast(policyId: string): OperationResult spilloud Pulicy(deptin : int) : Node spilloud Pulicy(deptin : int) : Node spilloud (red Element) di stating, deptin : int) : Node spilloud (red Element) di stating, deptin : int) : Node spilloud (red Element) : Node spilloud (red Element) : Node spilloud (red Element) : Node spilloud (red di stirring, int) deptin) : Node spilloud (red Element) XACMLNodeAdaptor +insertElement IntoPalicyAfterElement(policyId: string, elementId: string, element : Node): Operation Result +removeElementFromPolicyTreeElement(elementId: string, elementName: string): OperationResult +removePolicySet(policySetId : string): OperationResult. +removePolicy(policyld: string): OperationResult +removeRule(ruleId:string): OperationResult +policySetExist(policySetId : string) : boolean +policyExist(policyld : string): boolean ecuseoo. +ruleExist(ruleId : string) : boolean moveRule(ruleId : string) : OperationResult licySetExist(policySetId : string) : boolean nicyExist(policyld : string) : boolean leExist(ruleId : string) : boolean BDB XmlHelber ligações BDBXmlDatabaseHelper +addNamespace(prefix : string, namespace : string) : void addNode(nodes : string, where : string) : XmlResults +addNodeAfter(nodes:string,which:string):XmlResults +addNodeAsFirst(nodes : string, where : string) : XmlResults +addNodeAsLast(nodes : string, where : string) : XmlResults +dearNamespaces(): void <cuseoo +dase(): vaid createDocument(name:string,content:string): XmlDocument ts getEnvironment(): Environment getEnvironmentConfig(): EnvironmentConfig +getXmiContainer(): XmiContainer +getXmlManager(): XmlManager +getXmlManagerConfig(): XmlManagerConfig putDocument(uniqueName : string, fileName : string) : boolean odateNode/newl query(xQuery: string): XmlResults removeDocument(docName : string) : boolean removeNamespace(prefix : string) : void removeNade (which : string) : XmiResuits Inovação updateNode(newNodes:string,where:string):XmlResults Francisco Alexandre de Gouveia

Extensibilidade (Módulos implementados)



- Info Retreiver
 - Lê ficheiro xml
- Devolve:
 - Tipos de dados
 - Funções
 - Algoritmos combinatórios





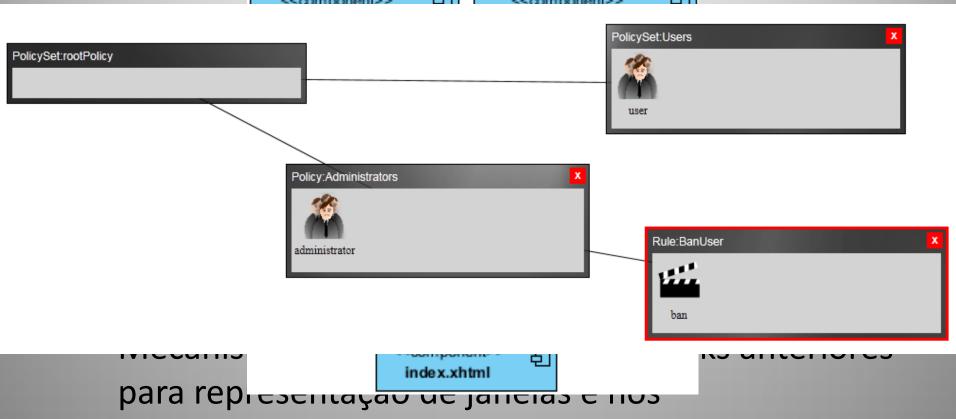
 Abstracção da complexidade do XACMLv3 na interface de administrador

- Solução usada:
 - Interface web com:
 - Representação de políticas em nós com ligações entre eles
 - Abstracção de nomes complexos através de imagens representativas e nomes simplificados
 - Abstracção das regras do XACMLv3 através de opções únicas para cada contexto





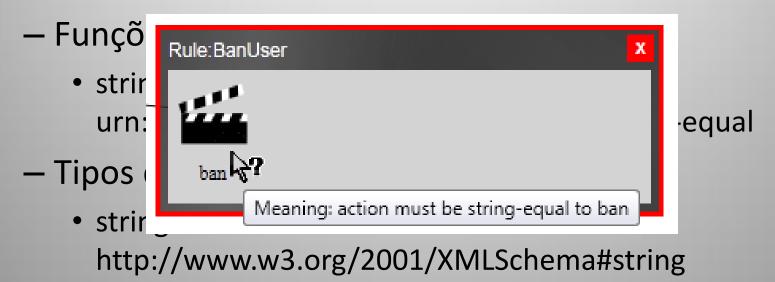
• Representação de políticas em nós com ligações







 Abstracção de nomes complexos através de imagens representativas e nomes simplificados



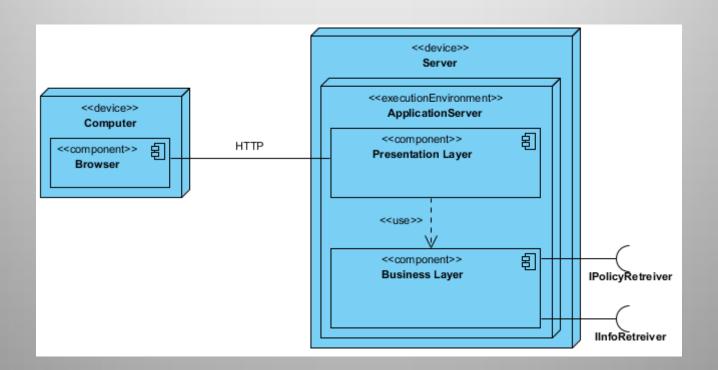




- Criação de políticas que obedeçam à norma
 - Existe um esforço para que, por trás de toda a abstracção, as políticas sejam criadas conforme a norma especificada
 - Em cada contexto, as opções mudam em relação às possibilidades de cada elemento
 - Por exemplo, existe uma toolbox que muda os botões dependendo dos elementos seleccionados



Implementação: aplicação J2EE





- Implementação: aplicação J2EE (problemas)
 - ClassLoader não funcionou quando colocado em funcionamento numa aplicação web
 - Classes de instância única não tinham sempre a mesma instância





Class Loader do Java

Funciona de modo hierárquico

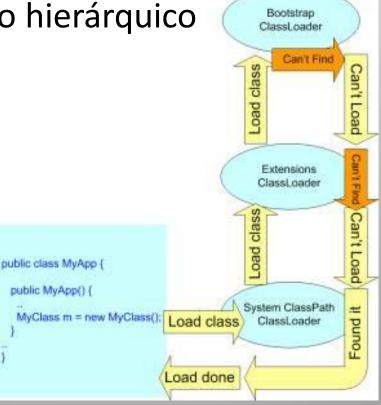


Imagem retirada de http://www.objectsource.com/j2eechapters/Ch21-ClassLoaders_and_J2EE.htm



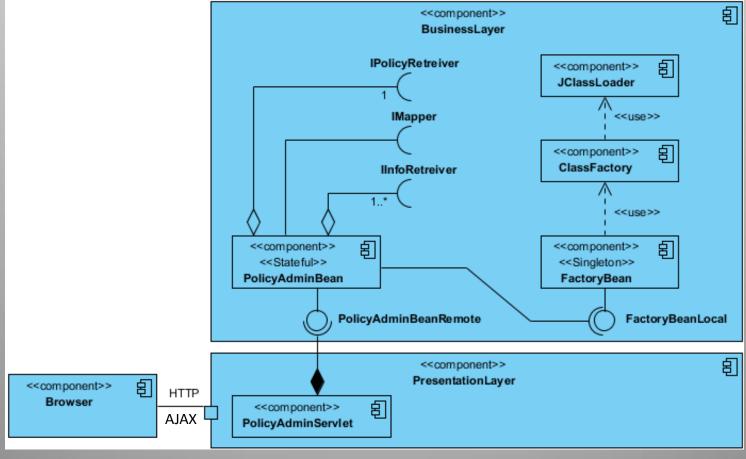




- Instâncias em web servers
 - Aplicação é colocada em mais do que um "contentor"
 - Para melhor performance, a carga é distribuída pelos contentores



Juntando tudo...









- Conclusões
 - XACML permite fazer controlo de acessos granulado e genérico
 - A versão 3 trouxe melhorias em relação à versão 2
 - Adição de Advices em alternativa a Obligations
 - Uniões e intersecções de Targets
 - Multi-request
 - Mas ainda não está terminada...
 - Não existem muitas implementações
 - Problemas de administração de Targets





Questões?



