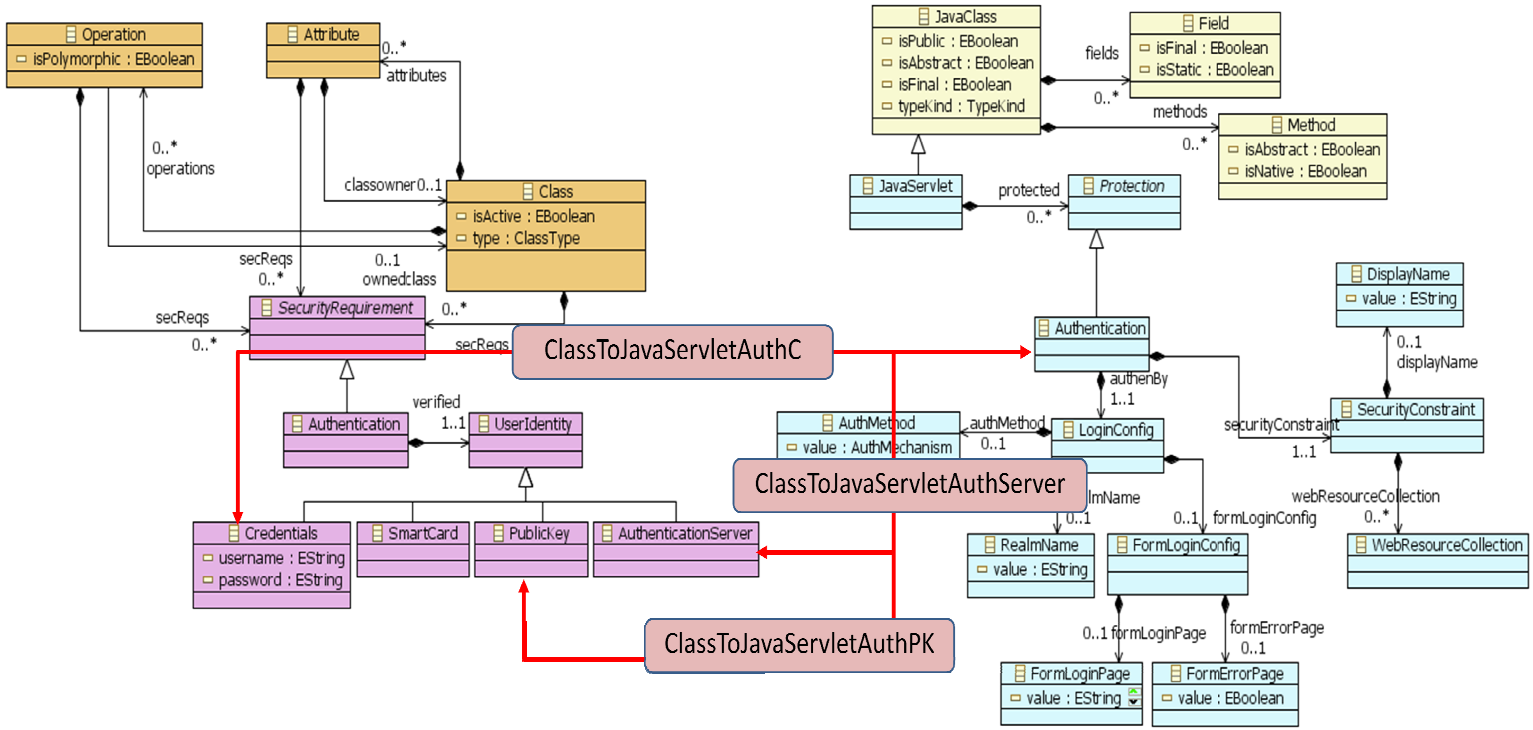
ภาพที่ 4.9

การจับคู่เมตาโมเดลของ PIM กับเมตาโมเดลของจาวาเซิร์ฟเล็ตใน

การรักษาความปลอดภัยเรื่องการพิสูจน์ตัวจริง



คำอธิบายกฎ ClassToJavaServletAuthC

กฎสำหรับใช้นิยามการแปลง Class ของเมตาโมเดลของแบบจำลอง PIM ที่มีการกำหนดความต้องการความปลอดภัยเรื่องการพิสูจน์ตัวจริงโดยใช้อิลิเมนต์ Authentication ที่ระบุประเภทของ UserIdentity แบบ Credentials ผลการแปลงแบบจำลองจะสร้างอิลิเมนต์ JavaServlet ที่มีอิลิเมนต์ที่เกี่ยวข้องกับการพิสูจน์ตัวจริงแทนด้วยอิลิเมนต์ Authentication ที่มีอิลิเมนต์สำคัญ 2 ส่วนคือส่วน LoginConfig คือการกำหนดวิธีการในการพิสูจน์ตัวจริง โดยในกรณีนี้จะใช้วิธีการแบบ BASIC คือการใช้มาตรฐานการพิสูจน์ตัวจริงของโปรโตคอล HTTP และส่วน SecurityConstraint คือกลไกในการกำหนดทรัพยากรภายใต้เว็บแอพพลิเคชั่น เช่นจาวาเซิร์ฟเล็ต เพจ JSP ที่ต้องการให้มีการพิสูจน์ตัวจริงก่อนที่จะเข้าใช้งานได้

ภาพที่ 4.10
กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการ
ความปลอดภัยในการพิสูจน์ตัวจริง (Credentials)

```
relation ClassToJavaServletAuthC {
          cn : String;
          isAbstract : Boolean;

          checkonly domain umlsec c:umlsec::Class{
            name = cn,
            isAbstract = isAbstract,
            type = umlsec::ClassType::service,
            secreqs = rq:umlsec::Authentication{
                  verified = user : umlsec::Credentials {
                        username = uname,
                        password = pass
                  }
            }
          };

          enforce domain jssec jc:javaserv::JavaServlet {
            name = cn,
            isAbstract = isAbstract,
            typeKind = javaserv::TypeKind::TypeClass,
            protected = prot:javaserv::Authentication{
                  loginConfig = l:javaserv::LoginConfig{
                        authMethod = auth:javaserv::AuthMethod{
                              value = javaserv::AuthMechanism::BASIC
                        },
                        realmName = r:javaserv::RealmName{
                              value = 'Default'
                        }
                  },
```

ภาพที่ 4.10 (ต่อ)

กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการ

ความปลอดภัยในการพิสูจน์ตัวจริง

```
                securityConstraint=scons:javaserv::
                      SecurityConstraint{
                      displayName = dsn : javaserv::DisplayName {
                            value = 'sec cons name '.concat(cn)
                      },
                      webResourceCollection=rc:javaserv::
                            WebResourceCollection{
                            webResourceName = wn :javaserv::
                                  WebResourceName {
                                  value = cn
                            },
                            urlPattern = up : javaserv::URLPattern{
                            value = '/'.concat(cn.firstToUpper())
                                        .concat('Servlet')
                            },
                            httpMethod = hm: javaserv::HttpMethod {
                                  value javaserv::HttpMethodValue::
                                        GET
                            },
                            httpMethod = hms :javaserv::HttpMethod{
                                  value=javaserv::HttpMethodValue::
                                        POST
                            }
                      },
                      authConstraint =ac:javaserv::AuthConstraint {
                            roleName = rn : javaserv::RoleName {
                                  value = 'manager'
                      },
                      description = d : javaserv::Description {
                            value = 'role description'
                      }
                      }
                },
                securityRole =  sr: javaserv::SecurityRole {
                      description = ds : javaserv::Description {
                            value = 'security role'
                      },
                      roleName = rname : javaserv::RoleName {
                            value = 'manager'
                      }
                }
             }
          };

          where {
             SecureAttributeToField(c, jc);
             SecureOperationToMethod(c, jc);
          }
}
```

คำอธิบายกฎ ClassToJavaServletAuthPK

กฎสำหรับใช้นิยามการแปลง Class ในเมตาโมเดลของแบบจำลอง PIM ที่มีการกำหนดความต้องการความปลอดภัยเรื่องการพิสูจน์ตัวจริงแทนด้วยอิลิเมนต์ Authentication และกำหนดประเภทของ UserIdentity แบบ PublicKey คือมีการใช้งานคีย์สาธารณะ (Public Key) การแปลงแบบจำลองจะสร้างอิลิเมนต์ JavaServlet ที่มีอิลิเมนต์ที่เกี่ยวข้องกับการพิสูจน์ตัวจริงที่มีโครงสร้างเหมือนกับกฎการแปลง ClassToJavaServletAuthC แต่แตกต่างตรงส่วนของประเภทของกลไกในการพิสูจน์ตัวจริงที่ใช้ ซึ่งในกฎนี้ กำหนดการใช้ AuthMechanism เป็นแบบ CLIENTCERT โดยกฎการแปลงแบบจำลองแสดงในภาพที่ 4.11

ภาพที่ 4.11
กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการ
ความปลอดภัยในการพิสูจน์ตัวจริง

```
relation ClassToJavaServletAuthPK {
          cn : String;
          isAbstract : Boolean;

          checkonly domain umlsec c:umlsec::Class{
             name = cn,
             isAbstract = isAbstract,
             type = umlsec::ClassType::service,
             secreqs = rq:umlsec::Authentication{
                   verified = user : umlsec::PublicKey {}
             }
          };

          enforce domain jssec jc:javaserv::JavaServlet {
             name = cn,
             isAbstract = isAbstract,
             typeKind = javaserv::TypeKind::TypeClass,
             protected = prot:javaserv::Authentication{
                   loginConfig = l:javaserv::LoginConfig{
                         authMethod = auth:javaserv::AuthMethod{
                               value = javaserv::
                                     AuthMechanism::CLIENTCERT
                         },
                         realmName = r:javaserv::RealmName{
                               value = 'Default'
                         }
                   },
          …
```

คำอธิบายกฎ ClassToJavaServletAuthServer

กฎสำหรับใช้นิยามการแปลง Class ในเมตาโมเดลของแบบจำลอง PIM ที่มีการกำหนดความต้องการความปลอดภัยเรื่องการพิสูจน์ตัวจริงโดยใช้อิลิเมนต์ Authentication และกำหนดประเภทของ UserIdentity แบบ AuthenticationServer คือมีการจัดการเรื่องการกำหนดผู้ใช้งานเอง ในการแปลงแบบจำลองจะสร้างอิลิเมนต์ JavaServlet ที่มีอิลิเมนต์ที่เกี่ยวข้องกับการพิสูจน์ตัวจริงที่มีโครงสร้างเหมือนกับกฎการแปลง ClassToJavaServletAuthC แต่แตกต่างกันในส่วนกลไกในการพิสูจน์ตัวจริงที่ใช้ ซึ่งในกฎนี้ กำหนดการใช้ AuthMechanism เป็นแบบ FORM ซึ่งกลไกแบบนี้ทำให้สามารถที่จะพัฒนาส่วนการตรวจสอบผู้ใช้งานได้เอง โดยกฎการแปลงแบบจำลองแสดงในภาพที่ 4.12

ภาพที่ 4.12
กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการความปลอดภัย
ในการพิสูจน์ตัวจริง (AuthenticationServer)

```
relation ClassToJavaServletAuthServer {
          cn : String;
          isAbstract : Boolean;

          checkonly domain umlsec c:umlsec::Class{
             name = cn,
             isAbstract = isAbstract,
             type = umlsec::ClassType::service,
             secreqs = rq:umlsec::Authentication{
                   verified = user : umlsec::AuthenticationServer {}
             }
          };

          enforce domain jssec jc:javaserv::JavaServlet {
             name = cn,
             isAbstract = isAbstract,
             typeKind = javaserv::TypeKind::TypeClass,
             protected = prot:javaserv::Authentication{
                   loginConfig = l:javaserv::LoginConfig{
                         authMethod = auth:javaserv::AuthMethod{
                               value = javaserv::
                                           AuthMechanism::FORM
                         },
                         realmName = r:javaserv::RealmName{
                               value = 'Default'
                         }
                   },
          …
```

2. กฎการแปลงแบบจำลองที่มีคุณสมบัติด้านความปลอดภัยในการรักษาความลับ

การจับคู่ระหว่างแบบจำลองต้นทางและแบบจำลองปลายทาง เพื่อนำไปนิยามกฎการแปลงเพื่อรองรับคุณสมบัติการรักษาความลับระหว่างเมตาโมเดลของแบบจำลอง PIM กับเมตาโมเดลของจาวาเซิร์ฟเล็ตกำหนดโดยใช้ชื่อกฎ ClassToJavaServletConf ซึ่งจะจับคู่ระหว่างอิลิเมนต์ Confidentiality ของแบบจำลอง PIM กับอิลิเมนต์ Confidentiality ของจาวาเซิร์ฟเล็ต ดังแสดงในตารางที่ 4.7 การสร้างกฎการแปลงเพื่อสร้างแบบจำลองปลายทางในส่วนการรักษาความลับจะสร้างอิลิเมนต์ SecurityConstraint และมีการกำหนด TransportGuarantee ให้มีค่าเป็น CONFIDENTIAL รายละเอียดของกฎแสดงในภาพที่ 4.13

ตารางที่ 4.7
การจับคู่อิลิเมนต์สำหรับจัดการความปลอดภัย
ด้านการรักษาความลับ

เมตาโมเดลของแบบจำลอง PIM	เมตาโมเดลจาวาเซิร์ฟเล็ต
Class: - ClassType=“service”	WebServiceClass
- Confidentiality	- Confidentiality
- Individual:AuthorizedParty	- WebResourceCollection
- Reading:Action	- UserDataConstraint
- PublicKey:UserIdentity	-TransportGuarantee= CONFIDENTIAL
- Confidentiality	- Confidentiality
- Individual:AuthorizedParty	-ไม่สามารถจับคู่กับอิลิเมนต์ใดๆ ได้
- Reading:Action	
- Credentials:UserIdentity	

ตารางที่ 4.7 (ต่อ)

การจับคู่อิลิเมนต์สำหรับจัดการความปลอดภัย

ด้านการรักษาความลับ

เมตาโมเดลของแบบจำลอง PIM	เมตาโมเดลจาวาเซิร์ฟเล็ต
- Confidentiality	- Confidentiality
- Individual:AuthorizedParty	-ไม่สามารถจับคู่กับอิลิเมนต์ใดๆ ได้
- Reading:Action	
- SmartCard:UserIdentity	
- Confidentiality	- Confidentiality
- Individual:AuthorizedParty	-ไม่สามารถจับคู่กับอิลิเมนต์ใดๆ ได้
- Reading:Action	
- AuthenticationServer: UserIdentity	