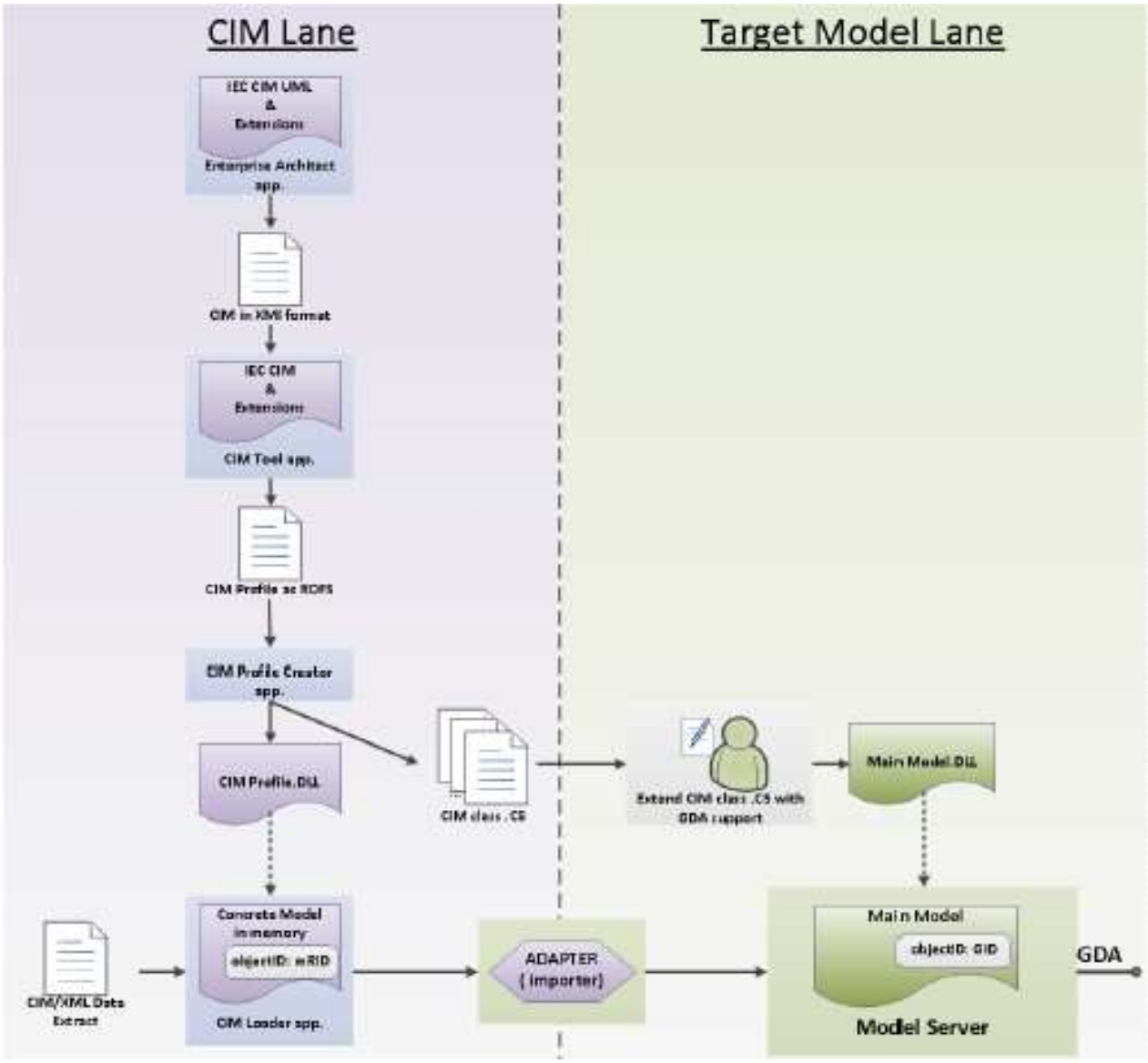
IES priprema za odbranu projekta

- ***CIM*** (Common Information Model) – model tj medjuanrodni standard IEC, opisan pomocu UML-a.
- ***RDF*** (Resource Description Framework) – definisan familijom World Wide Web Consortium, **opis mašinski čitljivih meta-podataka. Triplet (Subjekat,Predikat,Objekat)**.
- ***RDFS*** (Resource Description Framework Schema) – proširivi jezik koji obezbeđuje elemente za zadavanje RDF resursa.
- ***CIM Profile*** – skup definicija klasa, atributa i međusobnih veza u okviru jedne šeme. Naš profil je samo podskup od većeg nadređenog skupa klasa, tj. klase koje mi koristmo. Profil možemo zadati u više oblika, RDFS, XSD, HTML, SQL...
- ***CIM Tool alat*** – plugin za Eclipse platformu, dizajniramo CIM profil u njemu na bazi UML šeme iz XMI formata. Ovde smo pazili na kardinalitete, reference smo dodavali ali NE i listu referenci. Dodavali klasu pa zasebno njena polja i nakon toga osobine za polja.
- ***OWL*** – Web Ontology Language. The Web Ontology Language **OWL** is a language for defining ontologies on the Web. An OWL Ontology describes a domain in terms of classes, properties and individuals and may include rich descriptions of the characteristics of those objects.
- ***TOK PODATAKA PRI INICIJALIZACIJI MODELA ELEKTROENERGETSKE MREŽE*** - Prvi korak je Enterprise Architect i generisamnje CIM-a u XMI format. Nakon toga u CIM Tool generisemo RDFS. Onda u CIM Profile kreatoru konvertujemo RDFS u DLL. Ovo dalje je adapter koji je posle objasnjen.



- ***<u>Namespaces u RDFS</u>*** – primer :
  <rdf:RDF
  xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema# >

- ***<u>Deklaracija paketa u RDFS formatu</u>*** – primer:

<rdf:Description rdf:about="#Package_Wires"> - about je ID resursa <cims:belongsToCategory rdf:resource="#Package_IEC61970"/> - pripadnost drugom paketu

<rdfs:comment> …</rdfs:comment> <rdfs:label>Wires</rdfs:label>

<rdf:type rdf:resource="…" /> - tip resursa

</rdf:Description>

- ***Apstraktna klasa u RDFS formatu*** – npr. IdentifiedObject , ima about polje,belognsToCategory, type
- ***Konkretna klasa u RDFS formatu*** - … isto kao prethodne plus subClassOf(roditeljska klasa), stereotype(dodatni opis resursa)
- ***Primeri prostih klasa , tj Datatype klasa*** – razlika u odnosu na druge klase je sto ove ne sadrze belongsToCategory(podrazumeva se paket Domain), I subClassOf (nema generalizacije medju njima)
- ***Enumeracija u RDFS formatu*** – about, belongsToCategory, type, stereotype
- ***Vrednost enumeracije u RDFS formatu*** – bitni elementi about, type(referencira Enumeraciju putem ID-a)
- ***Atributi u RDFS formatu*** – about,type,stereotype, datatype(tip podatka, string, int, bool), domain(referencira klasu kojoj atribut pripada), multiplicity – kardinalitet , range referencira tip podataka koji je enumeracija

```
<rdf:Description rdf:about="#Package_Wires">
  <cims:belongsToCategory rdf:resource="#Package_IEC61970"/>
  <rdfs:comment>An extension to the Core and Topology package that models information
  on the electrical characteristics of Transmission and Distribution networks.
  This package is used by network applications such as State Estimation,
  Load Flow and Optimal Power Flow.</rdfs:comment>
  <rdfs:label>Wires</rdfs:label>
  <rdf:type rdf:resource="http://iec.ch/TC57/1999/rdf-schema-extensions-19990926#ClassCategory"/>
</rdf:Description>
```

- Primer *concrete* klase

```
<rdf:Description rdf:about="#PowerTransformer">
  <rdfs:subClassOf rdf:resource="#Equipment"/>
  <cims:belongsToCategory rdf:resource="#Package_Wires"/>
  <cims:stereotype rdf:resource="http://langdale.com.au/2005/UML#byreference"/>
  <cims:stereotype rdf:resource="http://langdale.com.au/2005/UML#concrete"/>
  <rdfs:comment>An electrical device consisting of  two or more coupled windings,
  with or without a magnetic core, for introducing mutual coupling between
  electric circuits...</rdfs:comment>
  <rdfs:label>PowerTransformer</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

- Primer apstraktne klase

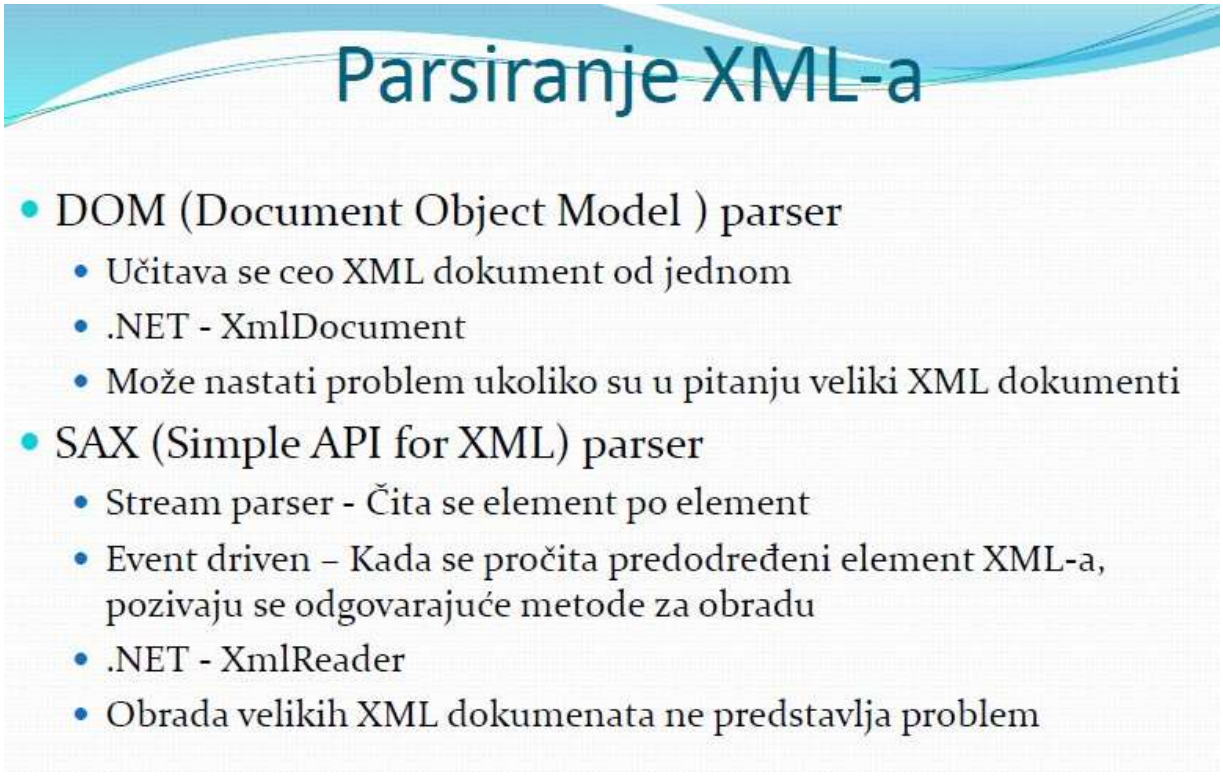
```
<rdf:Description rdf:about="#IdentifiedObject">
  <cims:belongsToCategory rdf:resource="#Package_Core"/>
  <rdfs:comment>This is a root class to provide common naming attributes for
  all classes needing naming attributes</rdfs:comment>
  <rdfs:label>IdentifiedObject</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

- Primer enumeracije

```
<rdf:Description rdf:about="#WindingType">
  <cims:belongsToCategory rdf:resource="#Package_Wires"/>
  <cims:stereotype rdf:resource="http://langdale.com.au/2005/UML#enumeration"/>
  <rdfs:comment>Winding type.</rdfs:comment>
  <rdfs:label>WindingType</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

- Primer vrednosti enumeracije

```
<rdf:Description rdf:about="#WindingType.primary">
  <rdfs:label>primary</rdfs:label>
  <rdf:type rdf:resource="#WindingType"/>
</rdf:Description>
```



- ***DMSType enumeracija*** – 16bitna enumeracija, svaka klasa cije se instanciranje ocekuje dobija odgovarajucu vrednost u DMSType enumeraciji (SAMO KONKRETNE KLASE IMAJU DMS TIP)
- ***ModelCode enumeracija*** – 64bitna enumeracija, 32 bita za nasledjivanje, DMSTip 16 bita, opis atributa 16 bita. Svaki tip resursa u modelu se jednoznacno identifikuje odgovarajucim ModelCode-om.

- ***ModelCode semantika*** – "Child" klasa uvek ima jednu cifru vise od "parent" klase. Ukoliko vise klasa nasledjuje roditelja, inda se cifra uvecava za svaku child klasu. Npr. IdentifiedObject nasledjuju PowerSysRsrs, BaseVoltage, a Equipment nasledjuje PSR , model kodovi ce biti IDOBJ 0x1000.., PSR 0x1100…, BaseVolt 0x1200…,Equip 0x11100..

- ***Pravljenje model kodova***: IDOBJ uvek dobije 0x1000000000000000 (ukupno 16 cifara sa desne strane) jer je na vrhu hijerarhije svih klasa. Prvih 8 cifara predstavlja nasledjivanje, drugih 8 cifara se deli na dms tip i atribut. Dms tip imaju samo konkretne klase. Atribut su poslednje 4 cifre i od njih prve 2 su redni broj atributa kako smo ga naveli kojim redom, a druge dve su bas tip npr. 07 za string , 04 za long(Int64)… I klasa i atributi imaju svoj ModelCode.

- ***Globalni identifikator(GID)*** – mRID jeste jedinstven za svaki entitet koji se kreira, njegova upotreba za identifikaciju entiteta moze da uspori rad servisa jer je u pitanju **STRING**. Zbog toga smo uveli **GID** koji je tipa **long** jer je poredjenje brojeva daleko brze od stringova. GID je definisan u klasi IdentifiedObject i zato svaka klasa koja se nalazi u hijerarhiji ispod ima globalni identifikator. GID je 64bitni. Kreira se na osnovu 3 podatka:

  1. 16bitni sistem ID(za nas 0 jer koristimo jedan sistem),
  2. DMSType odgovara tipu entiteta za koji se kreira GID,
  3. Brojac – za svaki tip entiteta postoji brojac koji obezbedjuje jedinstvenost GID-a po tipu.
- ***Reference – veze izmedju entiteta*** , modeluju se GID-om , reference su dvosmerne na servisu (ukoliko jedan entitet sadrzi GID drugog entiteta, onda I drugi entitet sadrzi identifikator prvog)
- ***GENERIC DATA ACCES (GDA)*** – omogucava pristup podacima bez ikakvog znanja o logickoj semi koja se koristi za unutrasnje skladistenje – implementaciji. Dovoljno je poznavanje CIM-a. Nadogradnjom kroz vreme, GDA obezbedjuje filtriranje, citanje i upis podataka. Ovaj standard formulise upite i odgovore u vidu resursa, propertija i vrednosti. Resurs je

svaki objekat koji poseduje jedinstven identifikator. Properti je aspekt resursa koji se moze opisati predstavljen preko ModelCode-a.

- ***Property*** – opisuje atribut nekog objekta, sastoji se iz dva dela a to su ID(svaki Property je jednoznacno odredjen preko ModelCode-a) i Value(vrednost atributa). Vrednost mozemo setovati kroz konstruktor ili preko SetValue metoda. Vrednost se cita preko AsBool, AsByte,AsInt,AsLong,AsString,AsReference …. metoda.
- ***ResourceDescription*** – opisuje objekat, moze da sadzri sve ili samo odabrane propertije. GID objekta koji se opisuje predstavlja identifikator ResourceDescription-a.
- ***Association*** – opisuje veze izmedju objekata, sadzri identifikator propertija tipa referenca. Vrednost propertija je GID referenciranog objekta.
- ***Metode za citanje podataka***:
  1. **GetValues**()–citanje jednog resursa,rezultat je ResourceDescription koji sadrzi prosledjeni identifikator I listu zahtevanih propertija. ***public ResourceDescription GetValues(long resourceId, List<ModelCode> propIds)***
  2. **GetExtentValues**() – citanje niza resursa za isti tip objekta ,metodi se prosledjuje ModelCode koji predstavlja tip entiteta I lista ModelCode-ova koji predstavljaju identifikatore propertija vezanih za entitet. Rezultat je identifikator upita na osnovu koga klijent moze da procita rezultujuce ResourceDescription-e.
     public int GetExtentValues(ModelCode code, List<ModelCode> propIds)
  3. **GetRelatedValues**() – citanje referenciranog resursa, Rezultat je identifikator upita na osnovu koga klijent procita rezultujuce RD-e. ***public int GetRelatedValues(long source,List<ModelCode> propIds,Association association) association je veza izmedju source I rezultujucih resusra.***
  4. ***GetDescendentValues***() – citanje niza referenci resursa , ***public int GetDescendentValuesLinq(List<long> sources, List<ModelCode> propId, List<Association> path, List<Association> tail)***
- ***Izmena(upis) podataka*** – **definise se objekat koji ce nositi potrebne izmene modela – Delta objekat**. Metoda ApplyUpdate() se poziva

prilikom izmene modela. Objekat tipa Delta sadzi tri tipa operacija : dodavanja, azuriranje i brisanje entiteta.

- ***Adapter*** – pretvara CIM-XML file u Delta objekat. Primenjuje Delta objekat na NMS(Network Model Service) kroz GDA .
- ***Metoda CreateDelta()*** – load process- extract i profil koristi za kreiranje ConcreteModel instance, transform process – na osnovu profila poziva odgovarajuci importer, importer transformise sadzraj ConcreteModel objekta u Delta objekat.
- ***Metoda ApplyDelta() ili ApplyUpdates()*** – koristi GDA interfejs ka NMS-u , pripremljen Delta objekat prosledjuje na primenu

- ***Importer*** je implementiran za odredjeni CIM profil. Ima znanje kako se elementi definisani u profilu mapiraju na objekte ciljanog DMS modela. Mapira CIM klasu na DMS klasu, CIM atribut na DMS atribut, ne mora mapiranje biti 1 na 1!

- ***CreateNMSDelta() –*** poziva konverziju . Konverzija kreira ResourceDescription instance na osnovu CIM objekta. Popunjavanje properija u okviru svakog ResourceDescription objekta. Bitan je redosled kojim se generisu GID-ovi zbog referenci.

- ***Import<Type>() metode:*** selektuju listu CIM objekata datog tipa iz ConcreteModel-a , za svaki objekat iz liste kreira se ResourceDescription instance i popunjava sa Property podacima. Prilikom transformacije u ResourceDescription potrebno je generisati gid I upamtiti mapiranje CIM identifikatora na GID -> ovo radi ImportHelper

- ***Konverter -***> mapira attribute CIM objekata na ModelCode I ispravno postavlja vrednost svakog propertija. Populate metoda .

- ***Network Model Service (NMS):*** Expose-uje WCF interfejs INetworkModelGDAContract na adresi: net.tcp://localhost:10000/NetworkModelService/GDA/. Podesavanja za binding I serviceBehavior su u konfig fajlu servisa.

- ***Common*** ucitava svaka aplikacija sistema, opisuje model koji postoji na NMS. ModelDefines definise sve model kodove I dms tipove. Enums.cs sadrzi definicije enumeracija naseg modela. EnumDescs mapira enumeracije na attribute odredjene klase. ModelResourcesDesc obezbedjuje metode za rad sa model kodovima na osnovu informacija koje su definisane u samoj vrednosti model koda. Ova klasa definise listu atributa CIJA VREDNOST NE SME DA SE POSTAVI OD STRANE KLIJENTA. NotSettable ona metoda u prevodu.(samo atributi tipa lista referenci)

- ***InitializeTypeIdsInInsertOrder() –*** definise redosled izvrsanjava operacija koje stignu na servis kao Delta objekat(insert, update, delete).Ovde se navode samo model kodovi konkretnih klasa. Metoda se nalazi u ModelResourcesDesc klasi.

- ***Klasa Container*** – grupise entitete istog tipa, mapira GID-ove na njihove entitete. Kreira I pirbavlja entitete koji ima odgovarajuci GID. Metoda CreateEntitiy() mora obezbediti kreiranje bilo koje konkretne klase na osnovu GID-a.

- ***DataModel projekat*** – sadrzi implementaciju svih klasa koje ulaze u nas model. Potrebno je za svaku klasu implementirati Equals i GetHashCode. Takodje treba implementirati i regione IAccess implementation i IReference implementation. Da bi servis radio na jedinstven nacin sa svakim tipom entiteta potrebno je da klasa implementira metode koje servis koristi za GDA manipulaciju(IAccess,metode Has, Set I Get Property). Drugi region sadrzi metode koje servis koristi kako bi samostalno mogao da manipulise atributima koji su tipa lista referenci(Metode IsReferenced, Get,Add,Remove reference.).

- ***HasProperty*** proverava da li tekuca klasa sadzri atribut koji odgovara model kodu koji se proverava. Proverava prvo za svoja polja a onda i od roditeljske klase.

- ***GetProperty()*** vrsi konverziju atributa klase u Property objekat koji koristi GDA standard. Mora obezbediti mogucnost citanja svakog atributa klase. Ako je atribut tipa enum mora se konvertovati u SHORT.

- ***SetProperty()*** – metoda sluzi za postavljanje vrednosti atributa na osnovu Property-a koji koristi GDA standard.

- ***Atribut klase koji predstavlja referencu*** je tipa LONG. Njegova vrednost je GID entiteta kog referenecira.

- ***Atribut klase koji predstavlja listu referenci*** je lista tipa long(List<long>) I njegova vrednost je lista GID-ova entiteta koji imaju reference na pomenutu klasu.

  Ovakve klase imaju metode koje omogucavaju servisu manipulaciju nad referencama.
- ***Equals()*** mora da proverava jednakost svih atributa pa i referenci. Koristimo CompareHelper.CompareLists za listu referenci.
- ***HasProperty()*** mora da ima I ove model kodove referenci. ***GetProperty()*** mora da omoguci i citanje ovih atributa.
- ***IsReferenced() –*** implementiraju samo klase koje imaju listu referenci kao neki od atributa. Vraca indikaciju da li postoji neki entitet koji referencira instance ove klase.
- ***GetReferences()*** kreira mapu koja vraca vrednost GID-ova koji su vrednosti tipa referenca ili lista referenci.
- ***AddReference() –*** koristi servis kako bi dodao odgovarajucu vrednost u atribut tipa lista referenci.
- ***RemoveReference()*** koristi servis kako bi uklonio odgovarajucu vrednost Iz atributa tipa lista referenci.