# Proposed Plan for JSON Task Force

1/22/18

## 1. Define Rules

The task force will complete its translation rules for creating a programmer friendly JSON from XML by minimizing any type checking (array vs. object vs. simple property value pair) and by making accessors (e.g., dot notation) in programming languages natural for the programmer. The task force will document these rules as a PESC and A4L standards.

## 2. Create an Executable Translator and Validator

The task force will create an open source Java Object implementation from the College Transcript Schema that implements the serialization of JSON from an XML instance document in accordance with the PESC/A4L translation rules standard. The task force will decide if this implementation also performs validation on both XML and JSON or if a separate program is required. (It is assumed that A4L will do a similar implementation with one of their standards.)

## 3. Demonstrate at Spring Summit

The task force will demonstrate the JSON generated from an XML instance document and show that it is valid through the validator.

## 4. Specify Object Access through Services

The task force will define the objects within the College Transcript that may be accessed through an API or define a query approach to get any object. The task force will then determine the access URLs, the request data and http operations (get, post, etc.) This will be documented in a Swagger document. In addition to access to transcript objects, the API would support translation and validation of instance documents.

## 5. Implement a Web Services Reference Model

Using Swagger generation for Java and the College Transcript Java Object Model, implement a web service to access objects in of a transcript in XML or JSON. The Swagger specification would be provided to the membership and allow them to unmarshall their own transcript and use the API to access its parts. This service could be set up as a translation service for full or partial transcripts.

## 6. Define the API Documentation for all PESC and A4L Standards

The PESC and A4L communities would define the APIs for accessing, validating, and translating all standards. The exchange of information would be defined as accessing a provider's end points to retrieve the needed information. This could These API specifications would become the standard and the XML standards would be deprecated over time.

## 7. Create Linked Data Model

Currently, PESC standards primarily use parent-child relationships to specify the relationship between objects. This usually denotes "is part of" or "is a property of" relationship of the child to the parent. This limits the semantics of PESC standards in specifying many to many relationships. For example, students have relationships with a multiple transcripts, and these may be distributed across many institutions. In addition, the information about that student is duplicated in multiple documents. However a receiver may wish to retrieve all of these transcripts. One approach is to define independent objects and link them together with the various relationship types. For example, there could be a relationship object between a student identifier and an academic course. The requester could then pull the student information and the course information from the API using the linked data in the relationship object. This model appears to be better suited for web services. This model is also closer to how A4L represents their objects.

## 8. Schedule

| Activity | Date | Comments |
|---|---|---|
| Define Rules | 2/12/2018 | |
| Create an Executable Translator and Validator | 3/5/2018 | |
| Demonstrate at Spring Summit | 4/3/2018 | |
| Specify Object Access through Services | | |
| Implement a Web Services Reference Model | | |
| Define the API Documentation for all PESC and A4L Standards | | |
| Create Linked Data Model | | |