# Thesis Proposal:
# Developing an ABAC based Grant Proposal Workflow Management System

Milson Munakami
milsonmunakami@u.boisestate.edu
Boise State University

May 2, 2016

# 1.Introduction

With the advancement of Web infrastructures and technologies such as Cloud Computing, Big Data, Bring Your Own Device (BYOD), Internet of Things(IoT), distributed systems and web services, etc., organizations are trying to adopt such new trends to develop and implement autonomous workflow management systems (WfMSs). By leveraging the benefits of such advanced and cutting-edge, innovative information technology, it is bringing a new paradigm shift in the organization breaking the traditional approach of manual paper-based workflow management. Primarily, such online WfMSs focus on helping people to perform their tasks better and faster. However, the same level of security and automation is required by the organization along with promoting collaboration and information sharing among its stakeholders. As such fast-paced business processes are automated commonly referred as 'workflow automation' many security challenges need to be considered to streamline the work associated with each process step to make it more secure and flexible. Such dynamic and adaptive WfMS needs to provide a way to adopt the vibrant and more changing organizational need both functional and security requirements. Additionally, it also needs to offer customers the ability to focus on work and improve business operations rather than managing and tackling new information security challenges associated with each task.

To accomplish security requirements of any adaptive workflows, we can implement access control mechanisms [1][2][3][4]. According to National Institute of Standards and Technology (NIST) - "An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions." [5]. In recent year, many secure access control models [2][5][6][7] have been proposed and studied for collaborative and intra-organizational environments that express complicated access control constraint using traditional security methods. Unfortunately, those old static access control models radically fail to meet new regulatory and compliance demand of a real-world organization. In particular, the majority of available workflow systems do not yet support external authorization. In these models, access is defined and controlled by each application's backend database or via hard-wiring within code-level which can make them harder to address the dynamic organizational changes and restructuring processes. Contemporary information

security mechanisms are often immature or insufficient in addressing such demanding compliances. To make such WfMS more secure and maintainable, we need to separate clearly the business logic from the security features so that authorization logics do not need to be managed within the code rather can be created and maintained external to the application.

Using autonomous workflow systems can leverage significant advantages to organizations by way of reduced paperwork and accelerated the flow of document-centric information through automatic electronic documentation routing and better Quality of Service (QoS) to their users. By adopting such digital transformation, we face several challenges such as automation and security managements alongside we need to take account of different access control constraints. In a workflow, Security involves the implementation of a secure access control mechanisms to ensure that no subjects are allowed to perform unauthorized activities.

In this research, we are going to implement security model with the separation of authorization in a real-world application such as Grant Proposal Workflow Management System (GPWfMS) in which we try to capture the real-world workflow process of University Grant Proposal Submission operation. GPWfMS is a web-based workflow management system to automate and regulate the approval process of grant proposal submission which manages the creation, routing, and processing of grant proposals necessary to complete a transaction. In this workflow system the authorization and administration permissions are distributed to multiple administrative domains and administrative roles. The traditional centralized authorization and policy models in access control have many limitations that cannot fulfill the regulatory requirements of dynamic and adaptive collaboration environments. Thus, there is a great need for flexibility in software design and implementation that supports dynamic changing of security policies based on complex access control features like Delegation of Rights or Authority (DOA) and Obligation security constraints. In our service-oriented web-based Grant Proposal management system, the example of the proposed access control model defining the business activities in the workflow life cycle is as shown Figure 1.

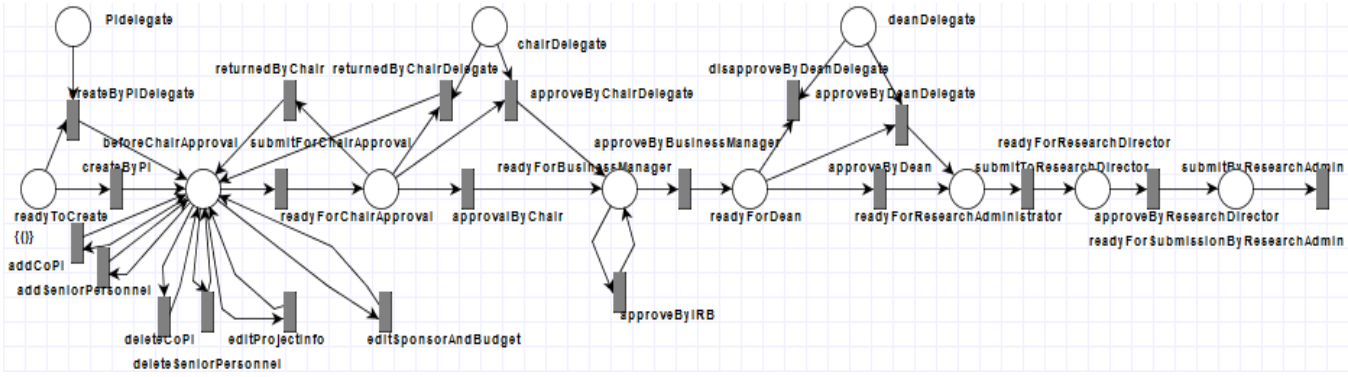



**Figure 1 Proposal workflow life cycle**

Currently, the process of creating a research proposal and routing it for final submission is a very time-consuming and manual process as any proposal may require multiple users to review and approve it during various stages. This extensive process begins filling up much sensitive information as shown in proposal data sheet in Appendix VI. Each workflow steps can act as an entry point for potential security

threats and attacks, such as any unauthorized access to the protected sensitive organizational information and leakage of user data. If such data is leaked, it can cause a security risk to the whole organization. The increasing interest in replacing paper-based workflow into the internet based online distributed workflow systems in multi-users and the multi-objects environment makes it vulnerable to security attacks and threats from outsiders. However, broad adoption of such system makes security, confidentiality and privacy issues more important to any organization using them as an integral part to manage their daily workflow. Such paradigm shift is increasing the complexity of workflow software architecture, design, and implementation. Hence, a more efficient and secure system design is needed to protect the immense flow of sensitive information flowing through such system from potential data theft. Improper design and implementation of such access control security constraints may arise critical complications. These restriction issues within a workflow system will be explored within the context of the execution.

The proposed model can easily adopt dynamic authorization to overcome the limitations of non-essential coupling between user/privileges and roles. This model provides a concept of user/permission pool based on the dynamic organizational structure. The system based on this model will be more secure and flexible because of its expressiveness to define complex access control policies. We proposed a bottom-up approach for more refined security based on attributes held by each user and resource in an organization. With Attribute-Based Access Control (ABAC), we can easily add any additional context using various attributes (i.e. Subject, Action, Resource, and environment or user defined, etc.) to any request while a user is trying to access a resource. The proposed logical model using ABAC with eXtensible Access Control Markup Language (XACML) highlights the importance of using both roles and hidden contextual information to make access strategy in the authorization process instead of single role information. Such fine-grained access control makes the system more secure and reliable.

# 2.Thesis Statement

The purpose of this research is to propose and create a more secure and reliable software design model that uses ABAC using XACML policy driven by administrative delegation and obligation rules and implement the model into a real world workflow management system. We propose a separation of code-based business logic and authorization policies by utilizing the XACML and making use of fine-grained access control model called ABAC. In particular, we are going to investigate various security concerns in a complex environment of GPWfMS, which captures the real-world working process of Grant Proposal Submission. This research allows us to examine into some of the advanced access control concepts like DOA and Obligation. Issues of DOA can cause a critical security threat to the business as it provisions more administrative authority to any new user in absence or consent of a primary delegator. Moreover, in current existing legacy workflow systems, there is no way we can enforce Obligations such as responsibilities or duties on any users. This allows us how can we properly implement such constraints into the system design and implementation phase and how can we properly handle the advanced access control concepts of DOA and pre/ post obligations within the business logic/ policy level rather than embedding into the source code level.

# 3.Motivation

A workflow involves a sequence of related tasks that are performed automatically to achieve an organizational goal [8]. In other words, a workflow can be described simply as the movement of documents and activities through a business process among different users. Today, workflow systems involved the automation of a business process involving more coordinated and collaborated execution of multiple tasks from different entities that may reside outside the inter-organizational boundaries at distributed environments. Such intra-boundaries access demands such system to support for continuous and collaborative business process improvements which put the business process immediately and directly under the control of the people using the system. With such improvements the problem of interoperability arises and to alleviate such problem; we need to adopt an efficient mechanism to establish trust among participants in a high-level abstraction.

Many Web-based workflow applications enhance their safety via access control systems [1][2][3]. Available state-of-art digital solutions have security access controls hardcoded within the code or do not specify the access control features such as DOA and Obligations in policy level. These limitations making such applications rigid, incomplete, less secure and easy target to the security threats. The primary focus of the security in such model is based on their role in the organization which can quickly restructure or revoked in dynamic enterprises which means again the code need to be reconfigured and modified which is more time and space consuming.

In the study of workflow secure access control models, the task-based access control (TBAC) and role-based access control (RBAC) are most commonly considered and applied [9]. The concept of role-based access control (RBAC) began with early multi-user, and multi-application online systems pioneered in the 1970s[10]. The traditional RBAC model is insufficient that cannot give fine-grained access constraints. RBAC imposes many limitations for the granularity of permissions among heterogeneous domains, resources, and users. From an enterprise perspective, RBAC is a passive access control model based on the direct assignment of roles and responsibilities that specify no time constraints, which can be exploited and can cause security threats. Such mechanism can be very messy and complicated if an organization has hundreds of thousands of users and corresponding roles that lead to “role explosion” problem. Also, revocation of users from assigned roles can cause another big problem to the organization administration. Changes to these associations between roles with privileges and users with roles are infrequent and explicit. Unable to do so can cause many unforeseen security risks and may not correctly reflect the business requirements.

NIST[11] indicates ABAC as a recommended access control model for promoting information sharing among diverse and disparate organizations. ABAC is a relatively new paradigm for handling security policies and access control. ABAC is more dynamic logical access control methodology where authorization for activities is determined by analyzing attributes associated with the subject, object, action and environment conditions against policy rules that define authorized operations by a subject on some resources. RBAC falls short of addressing dynamic fine-grained authorization at runtime. On the other hand, due to its fine-grained nature, ABAC can be used to facilitate secure information sharing

within the organization or intra-organization environment, without losing full control over it. Unlike RBAC in which job function (role or identity) of a particular user defines an authority level; ABAC facilitates collaborative policy administration and auditing. ABAC explains not only WHO can access WHAT but also provide some additional context like WHEN, WHERE, WHY, and HOW. In simple words, ABAC relies upon the matching of attributes of the subject, attributes of the object, environment conditions, and their relationship with defined access control rules. ABAC consists of all core features of other access control mechanisms such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Context-Based Access Control (CBAC), Task Based Access Control (TBAC) and RBAC, which makes ABAC backward compatible with the traditional access mechanisms and more diverse than others.

We propose a separation of code-based business logic and authorization policies by utilizing the XACML and making use of fine-grained access control model called ABAC. The XACML is XML-based declarative policy language for defining access control policies and a related processing model which permits the specification of authorizations as rules. Furthermore, XACML is a generic framework recognized by OASIS standard[1] for access control which ideally provides expressiveness, modularity, interoperability and efficiency [12][13]. The proposed security model combines the advantages of the new fine-grained ABAC model along with other security access control constraints such as DOA and Obligations. These capabilities offered by ABAC enable truly fine-grained and dynamic authorization that can be made context-aware and risk intelligent.

As shown in Figure 2, XACML Policy Language Model compose of many components. The main elements of the XACML Policy Language model are:

1. Policy Sets consists of one or more policies.
2. Policies: A policy includes a set of rules, a declaration for applicable rule-combining algorithms, a set of obligations and advice, and a target.
3. Rules: The most elementary unit of policy. Policy can comprise of one or many rules that can evaluate to *Permit, Deny, Indeterminate*, or *Not Applicable*.

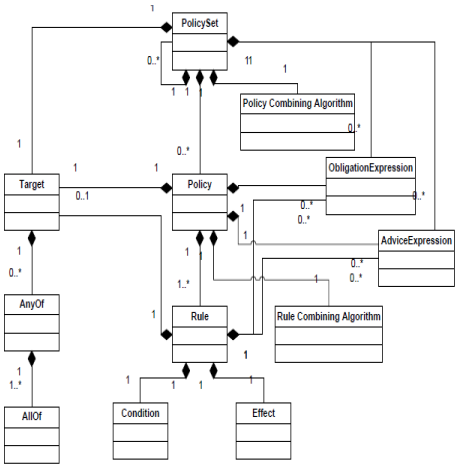



**Figure 2 XACML Policy Language Model**

Even though we are experiencing unprecedented popularity rise of WfMSs, very few efforts are done to take into account for access control constraints such as Separation of Duties (SoD), DOA and Obligations. The latest version XACML 3.0 has integrated obligations and also added generic attribute categories into the context and extended policy delegation profile also known as administrative policy profile. However, it does not specify what an obligation and delegation profile should include and how software design needs to handle them properly. Such immaturity in XACML is making these new access control concepts not widely applied yet as well as there not many examples are available. Also, there is very few related work has been done on the real use case for both specifying, dynamically enforcing and implementing access control policies taking into account such complex access control capabilities.

[1] https://www.oasis-open.org/standards

The complexity of real-world workflow application requirements is revealing limitations of the current security model design. The available traditional security access models are more discretionary and do not consider contextual information such as date time and environments that make intruders easy to bypass such limited security mechanisms. One of the many outstanding technical challenges of adaptive WfMS is that it need to unify people and resources with diverse features into a more cohesive way. Our goal of this research work is to improve the existing secure software design model that mainly advocates for the use of TBAC, RBAC [9] and ABAC without the concept of DOA and Obligations.

# 4.Methodology

The overall software design of the GPWfMS web-based application can be summarized as shown in Figure 3; that shows an abstract view or representation of core components. When creating workflow systems, we need to keep in mind about the coordination of activities, resources, data, and applications. To develop guidelines for the design of a workflow-enabled organization; we need to understand an overview of the business aspects of workflow technology in the context of the workflow life cycle.

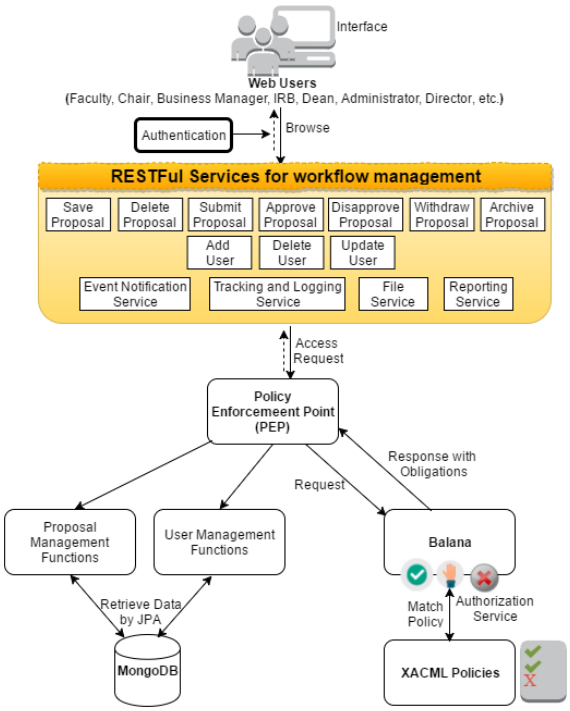



**Figure 3 GPWfMS Software Architecture**

## 4.1 Proposal Workflow

As we can see in Figure 1 and UML diagrams in Appendix I, II and III, a research grant proposal is written and initiated by a Principal Investigator (PI). The proposal may include some Co-Principal Investigators (Co-PI) and Senior Personnel as co-authors or contributors. When the PI is ready for the proposal to be evaluated, she can submit it to a department chair for approval who will either return it or route it to the next phase in the workflow where it will await for being reviewed by the Business Manager, the Institutional Review Board (IRB) and the Dean. This process can get even more complex and complicated if

the proposal involves investigators from multiple departments as all departments' authoritative personnel need to review and approve the proposal content. Once the proposal is approved by the Dean as well as reviewed by IRB if it involves any compliance issues, then it must be routed to the University Research Administrator who can disapprove or withdraw it or can approve it to be routed to University Research Director. Research Director can either refuse or delete the whole proposal or can give final approval for submission by the Research Administrator.

As we can see in above described standard scenario, it involves different activities that need users with various position titles and privileges to engage and complete various tasks. Each activity within the workflow is associated with a subject who needs to ensure the pending work is completed on time, and all obligations are fulfilled before and after any action is performed. To convert the manual process into a flexible, reliable and more secure digital automated system is a challenge which respects the integrity of the workflow as shown in Figure 1. We can view this complex workflow as a multi-layered state machine which needs to fulfill pre-conditions and post-conditions in each state and some specific event triggers it to transition from one state to another. In particular, we are looking into a complex environment of GPWfMS which may involve various subjects trying to perform certain actions on the same resource that can alter data and control flow. Thus, it requires verification and validation of the correct access to the resource using subject's access levels which can be determined by corresponding subject and resource's attributes. Attributes may be considered characteristics of entities that may be predefined and pre-assigned a value by an authority.

Whenever a user with different attributes such as position titles (Faculty, Chair, Dean, etc.) and environmental context (e.g. Date, The polic repoTime, IP address) tries to enter the system via login web interface, he is authenticated based on his username and password saved in the backend database. We have used a level of security enforcement in this process, as the password is encrypted and made hard for guessing and brute-force. Once an authenticated user is logged into the system, he is allowed to view his pending proposal work items and also he is authorized to create a new proposal based on predefined the access control rules.

Software applications especially popular web application are using open web services i.e. web Application Programming Interfaces (APIs[2]) and using such public authorization service endpoint provides more interoperability among many distributed systems. In GPWfMS, APIs are gateways to connect enforcement points which control access to information from outside organizational boundaries. Such exposed web APIs allow any external applications to call the services of a workflow engine from outside the organizational boundaries. This leverages the work can be done outside the organization in federated environments. GPWfMS implements RESTful web services (JAX-RS[3]) based on JAVA Representational State Transfer (REST[4]) APIs to interact with the system and back-end database records.

[2] https://en.wikipedia.org/wiki/Application_programming_interface
[3] https://jcp.org/en/jsr/detail?id=339
[4] https://en.wikipedia.org/wiki/Representational_state_transfer

In particular, GPWfMS has four different services such as Proposal, User, File and Notification services that include various functionalities. Overall, proposal management system involves a sequence of activities from the creation of a research proposal to the final approval which is a very time-consuming and user-centric process as it requires multiple parties to review and approve in every step as shown in Appendix I, II and III. At each stage, the application must account for all actions that can occur from user interaction. In general consideration, these actions fall under a simple Create, Read, Update and Delete (CRUD[5]) paradigm. However, in the software design, our system must also account for the access control policies for each of these actions. Here we need to map properly Resources of Policy Model to business logic Resource Components. To evaluate and test the working of each service and functionality we have used jUnit[6] test cases.

**Proposal Management Services** includes i.e. Save, Delete, Submit, Approve, Disapprove, Withdraw, and Archive a proposal based on user credentials and access control. **User Management Services** provides functionalities like Add, Delete and Update a User. Besides. **Event Notification Service** is used to send notifications and alerts to all relevant users based on workflow status about changes and pending task lists. These functionalities are accessed by the administrator of the system. During each step, information regarding 'Tracking and Logging of Activities' are recorded logged onto system audit logs using **Tracking and Logging Service** to support non-repudiation. **File Service** provides an interface to upload and download files from the web server so that it can be used for 'Process Reporting' functionality. This service allows to monitor the currently available proposal items in the system and allows the user to create reports that display detailed information on current workload, future workload, obstructions, etc. based on "historic" processing data.

However, one of the critical issue while using such publicly visible services is security. Any unwanted hackers can expose user privacy and can do unauthenticated works via those public services. This is why to make them more secure and reliable we need to enforce access control associated with obligations, advice and delegation rules according to the context of the user request. To identify the participants and their associated privileges as well as to properly handle the proposal routing in each step ABAC based authentication and authorization mechanisms with advanced access control concept of DOA and Obligations is used. Such strengthen features going to be time-saving for involving people as well as going to make the software more secure and efficient.

## 4.2 Database Management

Contained within this system is a database that stores relevant subject's information. To manage the attributes of every subject and object, they must have corresponding entries in a database that allows attribute retrieval and comparison. The proposed dynamic architectural solution demands to generate, storing and analyzing more information with increasing speed and scale so to overcome such data-driven requirements we choose MongoDB[7] as the best suited No-SQL backend database.

[5] https://en.wikipedia.org/wiki/Create,_read,_update_and_delete
[6] http://junit.org/junit4/
[7] https://www.mongodb.org/

### 4.2.1 MongoDB

As shown in Figure 4, traditional 'relational' database model stores information in hierarchical rows and columns in a tabular format. However, such mappings and relationships are impossible in big and messy dataset harvested from vast and concurrent data streams that are evident in such workflow system. Besides, MongoDB is more Document-Oriented where each document is stored as JSON objects and stored as Attribute-value pairs that make it easy to retrieve and process data and also more human readable and scalable.

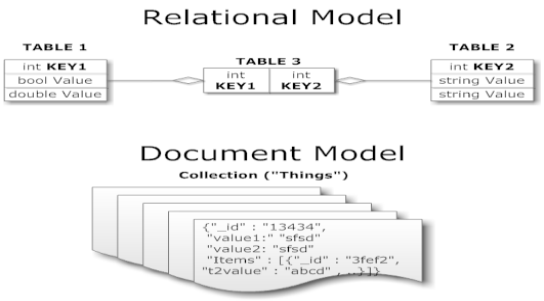



**Figure 4 Relational vs. Document Database Model**

Three primary database collections are needed for our purpose *Proposals*, *Users,* and *Notifications*. The User database collection contains the detailed information of a user as well as information necessary to authenticate securely access him with the system.

User information includes the following data:

- *User account data*: A user account name and password.
- *User detail information*: A user's given names, contact information (such as addresses, phone numbers, and email addresses), and departmental position/ role information.

The Proposal database collection contains information for a proposal including personnel information related to it. The information of proposal includes:

- *Project information*: Proposal specific information, such as the project type, title, date related information.
- *Financial information*: Budget details, sponsorship information, cost sharing information.
- *Investigator information*: Details about PI's, Co-PI's, senior personnel
- *Signature information*: Signatures and notes from corresponding authorized users.

The Notification database collection stores information regarding the recent changes to the data and the corresponding user whom the system need to notify.

### 4.2.2 Morphia

Morphia[8] is a lightweight library for mapping Java objects to and from MongoDB database. Morphia is an Open Source Fluent Query API that uses annotations and standards to interact with code and database. It adds a layer of abstraction between Datastore and Data Access Object (DAO) from the application. It eases the working with data in Java as it creates a data persistence interface in between. Morphia is MongoDB's Java Persistence API (JPA[9]) which is an object-relational mapping framework that solves the object-relational impedance mismatch problems and handles data access operations with less code by replacing direct persistence-related database accesses with high-level object handling functions.

[8] https://github.com/mongodb/morphia
[9] https://en.wikipedia.org/wiki/Java_Persistence_API

## 4.3 Access Control Policies

We have used XACML that is a declarative language for the specification of authorizations as rules. To maintain proper authorization between different users and resources, we will design and implement a series of XACML policies as shown in Appendix IV and V. The policies are defined and written by specified business rules and guidelines for access control to the system. XACML supports a variety of underlying infrastructures for policy and attribute storage. The rules, policies, and policy sets are stored in the policy repository that is accessible for access control. XACML standards address and define how security authorization requests are handled internally. Apart from processing the authorization requests and it also defines the mechanism to perform a complete analysis of rules, policies, and policy sets to come up with a correct decision. To make it work seamlessly through dynamic changes to its run-time environment, react and adapt to the rapid changes in process execution.

An example of an XACML rule as in GPWfMS, can be expressed and represented in the human readable format as shown in Figure 5 and can be read as:

*Any Tenured/ tenure-track faculty or Non- tenure-track faculty can add/create a new proposal only from internal network of the organization i.e. campus network. The subject attributes include 'Tenured/ tenure-track faculty'; the action attribute is 'Add' and the resource attribute is 'Whole Proposal' and environment attribute is 'Campus Network'.*

```
<Policy rule_combining_algorithm="deny-overrides">
  <Target />
  <Rule Effect="Permit">
    <Description>
        Only "Tenured/tenure-track faculty"
        or "Non-tenure-track faculty"
        can "Add" a New "Whole Proposal"
        from internal campus network
    </Description>
    <Target>
    <AnyOf>
        <AllOf>
          <Match>Tenured/tenure-track faculty</Match>
          <Match>Whole Proposal</Match>
          <Match>Campus Network</Match>
          <Match>Add</Match>
        </AllOf>
        <AllOf>
          <Match>Non-tenure-track research faculty</Match>
          <Match>Whole Proposal</Match>
          <Match>Campus Network</Match>
          <Match>Add</Match>
        </AllOf>
    </AnyOf>
    </Target>
  </Rule>
</Policy>
```

**Figure 5 Basic XACML Policy Format**

This rule stipulates that the request of a Tenured/ tenure-track faculty or Non-tenure-track faculty to create a new proposal will be granted only when he is doing so from within Campus network.

For proper proposal routing and real-time decisions making, we need to adopt dynamic, contextual, policy driven mechanisms. The policy driven nature of the decisions requires that the decision-making capability is externalized from systems/applications/services and not be embedded within the code. This externalized authorization helps to separate Database and Web Services access functionalities from security business rules for robust and flexible software design. On the other hand, using centralized security policies and mechanisms eliminates the tedious, repetitive, and labor-intensive manual procedures required to provision and manage security measures.

Figure 6 shows how each action in the policy can be mapped to the Java method in the business logic. Security Designer can update and modify the existing policy based on the business rules without concerning about tracing the source code. Access level rules are created and evaluated to determine how proposal-related information is controlled, processed, routed, and tracked to make smarter decisions in every activity.

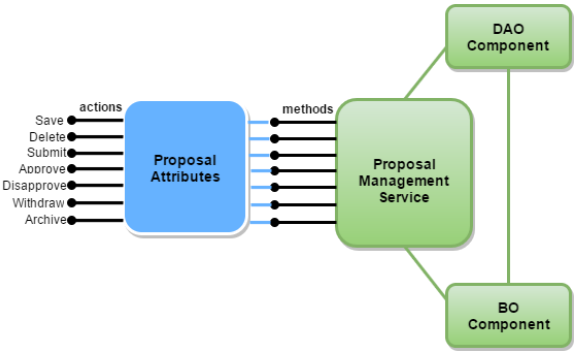



**Figure 6 Mapping Policy Model to BL Components**

The application holds the policy repository source folder in which we create a number of policy rules as shown in Figure 5 that directly maps the web services discussed in section 4.1. The final decision is based

on information about the subject, resource, environmental, and more hidden contextual information in the user request, that are often expressed as attributes and their corresponding values.

- **Subject**: position.title, position.type, proposal.role (PI, Co-PI, senior personnel).
- **Action**: Save, Update, Delete, Submit, Approve, Disapprove, Withdraw, and Archive
- **Resource**: proposal.section or whole proposal, proposal.status
- **Environment**: campus network, IP address, date time, mobile devices

Each access control rule consists of a condition, an effect, and a target as shown in Figure 5.

- Conditions are statements about attributes that can evaluate either True, False or Indeterminate.
- The effect return value Permit or Deny based on the satisfied rule.
- Target in policy helps in determining whether or not a rule is relevant for a request.
- As a policy can have multiple rules, it is evident that it can generate conflicting decisions based on different conflicting rules. To minimize that risk Rule-combining algorithms are used which resolve such conflicts and always try to outcome only one decision per policy.

However, XACML policy files include all rules/policies for the application which can have great security implications. Simple mistakes while writing wrong policy rules can grant unauthorized access and deny legitimate access to the system. Restrictive authorization and administration can be handled by the implementation of XACML security policies based on attributes; that can establish who can view, edit, and authorize specific parts of the proposal. An attribute is a property of an object; an authorization credential is a statement or assertion about an attribute. In particular, a credential must be based on defined attributes for a subject and during each action which validates and matches the pre-defined policy constraints.

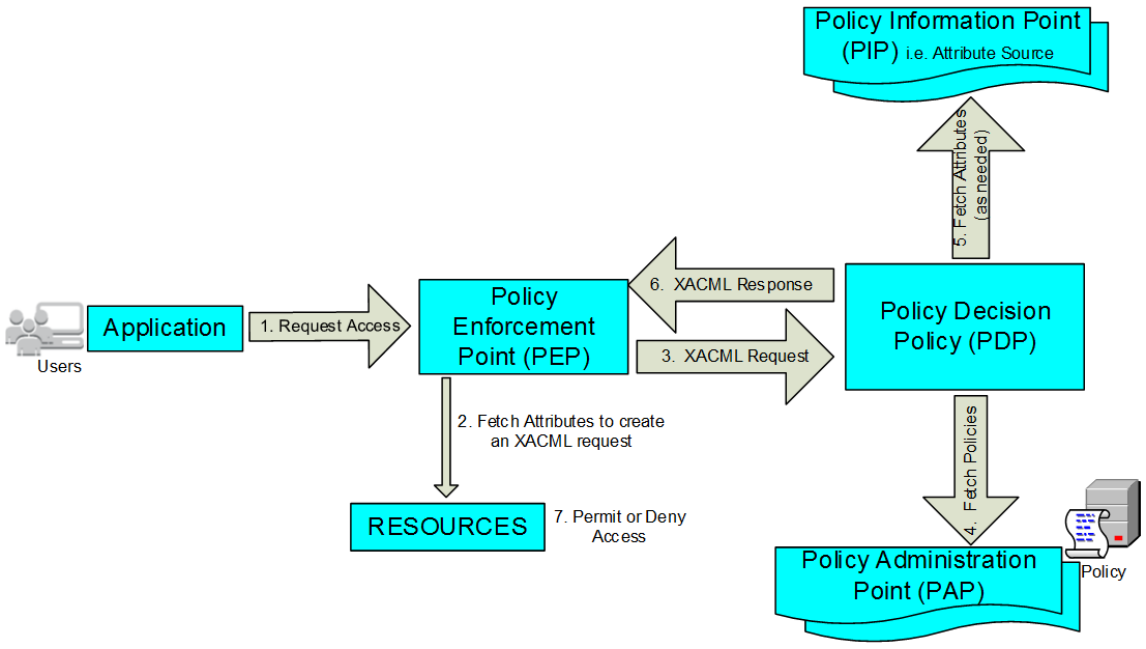



**Figure 7 GPWfMS with XACML Reference Architecture**

The challenge is to develop a good software architecture that can support complex security requirements which are common in the real-world dynamic organization. The architected solution can be viewed as interactions of three top-level components. The reference architecture as shown in Figure 7, depicts all the logical components of XACML reference architecture and their internal interactions with the application. The movement of data in our access control system will be as follow:

1. The system interface allows the user to log into the system by providing a verified username and his/her password as the first step. If the user is new to the system, he/she has to fill out the registration form to sign up for his/her account and the system administrator will have to activate the registered user by designating the correct position details.
2. When the user types his/her username and password, this information will be moved to the Policy Enforcement Point (PEP) which is responsible for receiving this information and fetching its attributes.
3. PEP also has to use these attributes to make an XACML request.
4. After PEP makes XACML request, this request for access or authorization decision will be forwarded to the XACML context handler with a predefined format that specifies the details about the attributes of the subjects, resources, actions, and the environment. The Policy Decision Point (PDP) analyzes the resource access request with the matching rules, policies, and policy sets. PDP has to fetch this XACML request and contact all available policies inside the system repository. The policy designer or administrator defines and manages policies and policy sets at the Policy Administration Point (PAP) based on current business rules.
5. PDP has also to fetch the attributes of sources from the Policy Information Point (PIP) if needed which behaves as a metadata of attribute values (i.e. a resource, subject, environment conditions). All these actions are made to create an XACML response.
6. After PDP makes XACML response, this replied response will be moved to PEP again which can read it and
7. Permit or Deny, the user to access his/her proposals.

The proposed software design and architecture makes the authorization mechanism more flexible and useful which simplifies the task complexity of security administrator. Currently, we are using Balana[10] which is an open source Sun's XACML Implementation by WSO2[11] that supports XACML 3.0 that allows creating Policy Decision Point instance in our web application. In GPWfMS, A request for authorization lands at the PEP. The PEP formats an XACML correct request using the context of the proposal data and then sends it to the PDP, which evaluates the request and sends back the decision in response. The decision response can either allow the access request or deny it. While making it more automated and web based we need to consider the possibility of having many 'disconnected users' who can obstruct the flow of the task. For handling such undesired but potential conditions, we need to refine the response and allow it to return all associated obligations and advice. To achieve this mission, we might need to extend the available XACML 3.0 standard specification to support more dynamic and robust obligations and delegation of authorities. We are proposing a software design model which implements ABAC along with advanced

[10] http://xacmlinfo.org/category/balana/
[11] http://wso2.com/

access control concepts such as DOA and Obligations (pre and post obligations) using XACML 3.0 to enhance security and reliability of such workflow based application that helps to model much closer to realistic business authorization scenarios. This new concept of constrained tasks to be followed before or after a request makes the software more secure and user more accountable and responsible.

### 4.3.1 Obligation

In modern WfMS, there is a need to have a connection between individual rights and embedded responsibility or between privileges and associated obligations. The obligation is very useful to solve responsibility and accountability problem in Structured and Collaborative environment. We seek to implement XACML 3.0's ability to fulfill an obligation which can be modeled as like pre and post conditions on each step of the workflow multi-layered state machine. Latest Obligation Specification that is extended in XACML 3.0 defines that each definition of the obligation contains a unique identifier and contains zero or many lists of parameters, each with a locally unique name and the data type as shown in Figure 8.

```
<ObligationExpressions>
  <ObligationExpression FulfillOn="Permit">
   <AttributeAssignmentExpression AttributeId="obligationType">
     <AttributeValue>preobligation</AttributeValue>
   </AttributeAssignmentExpression>
   <AttributeAssignmentExpression>
     <AttributeSelector Path="//ak:signedByCurrentUser/text()"/>
   </AttributeAssignmentExpression>
  </ObligationExpression>
  <ObligationExpression FulfillOn="Permit">
   <AttributeAssignmentExpression AttributeId="obligationType">
     <AttributeValue>postobligation</AttributeValue>
   </AttributeAssignmentExpression>
   <AttributeAssignmentExpression>
     <AttributeValue>Hello User, proposal updated by:</AttributeValue>
   </AttributeAssignmentExpression>
    <AttributeAssignmentExpression>
     <AttributeSelector Path="//ak:authorprofile/ak:fullname/text()"/>
   </AttributeAssignmentExpression>
   <AttributeAssignmentExpression>
     <AttributeSelector Path="//ak:pi/ak:workemail/text()"/>
   </AttributeAssignmentExpression>
  </ObligationExpression>
  <ObligationExpression FulfillOn="Deny">
    <AttributeAssignmentExpression>
     <AttributeValue>Invalid access by: </AttributeValue>
    </AttributeAssignmentExpression>
    <AttributeAssignmentExpression>
     <AttributeSelector Path="//ak:authorprofile/ak:fullname/text()"/>
    </AttributeAssignmentExpression>
</ObligationExpression>
```

**Figure 8 Obligations in XACML Rule**

XACML 3.0 allows us to describe an obligation method and its parameters as an attribute assignment so the actual definition of its syntax and semantics can be implemented quickly. Even though, the XACML policy language is very flexible; there is currently no generic method to specify the obligations send from PDP to PEP. It has no standard conceptual model for obligations and their enforcement [14]. Although, this is an important issue especially to support privacy, and advanced tracking of data flow this is quite neglected and not properly handled by XACML. Currently, the XACML standard does not provide a way to examine further the enforcement of an obligation and neither check its consistency and accuracy. Also, there is no concept of pre and post obligations. But in this research, we are going to extend the XACML 3.0 Obligation notion and make it support both pre and post obligations associated with each action.

Depending on the nature of the obligation, an obligation could be seen as an additional restriction on the access right that enrich the authorization flow. An XACML obligation is an action to be performed when a particular event is triggered. As shown in Appendix IV, a simple security rule having both pre and post obligations can be described as "Department Chair" can "Approve" a "Whole Proposal" when the current proposal status is ApprovedByDepartmentChair = READYFORAPPROVAL such that the **PRE** obligation is that the Chair needs to Sign it beforehand and the **POST** obligation is to Send Email to all proposal investigators to notify that the proposal is updated by him/ her". To specify such obligations in access control policies is more secure and flexible than hard-wired in code-level. We can implement Obligations in response to each user or program actions. Obligation statements define commitment or promise made by one entity to another entity. A list of obligations which have the same effect associated with an action is evaluated and returned to the PEP. Obviously, there can have conflicts among a set of obligations that means we need to keep account of relations between obligations to make it more accurate. The PEP is responsible for decoding and checking for each response for any obligations constraints and negotiates to enforce these obligations. For this, PEP keeps track of the obligations' state and enforce the restrictions.

For instance, a mere post obligation can be to send a notification email when a proposal is ready to be reviewed by the next person in the workflow. But before execution of the defined post obligation, she must sign the proposal; this defines the pre-obligation scenario. This kind of cases can be handled for both Permit and Deny access control from policy rules as shown in Appendix V. As shown in Appendix VI, the user action initiates a request that may have <*AttributeSelector*> element which may contain an XPath expression over the <*Content*> element of the resource to identify data in the information resource to be used in the policy evaluation. This kind of contextual information can be used by the Pre and Post obligation constraints. In this proposed model, we gather all the pre and post obligations from the request returned by matching response as shown in Appendix VII. Then the returned pre and post obligations are sorted based on their classification either pre or post. This classification is done based on the Attribute identifier defined as **obligationType** i.e. its value can be *postobligation* or *preobligation*. Based on the decision either Deny or Permit, the pre-obligations are checked first and validated that all pre obligations are satisfied. If not the precondition obligation will return the corresponding obligation response and can alert or show appropriate message to the user to fulfil the preconditions first. If all preconditions are satisfied and validated then the post obligations are executed in order. Our obligation strategy could supports multiple obligations within the same decided effect of the rule.

This process can be done autonomously by the system so it can lessen the workload for a user involved and assure less intervention is needed to manage the business process and can improve application performance and stability, also software maintainability. This ability to configure the obligation fulfillment requirement externally enables an administrator to activate or deactivate such security requirements dynamically without restarting or redeploying the running service.

### 4.3.2 Delegation

Apparently, the delegation of a task to another authorized user is a very useful real-world situation by which workflow continues to successful completion either in the case of user or resource unavailability or overloaded with tasks. The delegation of authority is an important business rule in an enterprise or organization where diverse users need to perform dynamic business processes in a heterogeneous computing environment. The Delegation of authority or rights allows a user, called the 'delegator', to delegate his/her access rights to another user, called the 'delegatee' [15]. Without DOA tasks cannot be divided among users so users would soon become overloaded with pending tasks. For example, considering our use case application GPWfMS, *the Department Chair may have to leave for a business trip; his approval work should be done by an Assistant Department Chair with whom he trusts. Therefore, he needs to give the person his permission to carry out the necessary task. Also, he and/or some specific authorized users is allowed revoke this temporarily delegated rights from his assistant once he comes back or anytime he wants.* Revocation can be done either manually or automatically by simply deleting the corresponding delegation rule. If such feature is not implemented in the workflow system, this can delay the overall task executions and can easily violate time constraints on the workflow impairing successful completion of the workflow. A delegation of Authority is a suitable approach to handling such exceptions and to ensure alternative execution routing path to the workflow process that makes WfMS more flexible and efficient. However, on the other hand, this decentralization in authorization can impose severe security risks to the organization by exposing high-level privileges to individual users. The basic idea behind

DOA is that an authorized entity is allowed to forward his authority to another active object so that later can carry out some tasks on behalf of the former.

In the field of access control, it is very crucial to have a delegation that helps to simplify the administrator task and to coordinate collaborative work securely, especially with the increase in shared information and distributed systems. Delegation and Revocation are important concepts that are essential for modeling and reasoning about dynamic distributed systems. Delegation is an essential and desirable feature in any modern enterprise and to model that constraint into a real-world software is a challenge because it brings lots of complexities, risk and privacy issues associated with individual user's privileges and permissions. Trust gives a notion of achieving such security constraints [16]. If this trust level is exploited, then that can be the point of security attacks and poses a threat to whole business.

In RBAC, it demands a significant number of delegation be created and managed as the number of roles and resources increase. However, this can be minimized by using ABAC model and also reduce the complexity of security administration. Delegation is a powerful feature for any dynamic business that provides a useful way to perform policy administration. On the other hand, it is necessary to monitor and make sure none of the security constraints are violated. Assignment of delegation can be based on time, workload and users' attributes. The dynamic and decentralized delegation distributes the privileges that make the workflow more flexible and scalable. Authority is often granted to an alternative subject if the primary subject is absent for an extended amount of time, and someone must be available to act on former's behalf. This situation typically occurs if there are not enough users to process the workload or user wants to offload his increased tasks with his sub-coordinates. At such situation, it is necessary to add additional resources to the workflow system. Delegation allows global administrators to delegate constrained administrative rights to local administrators. Thus, by dynamic delegation workflow system offers the user an ability to change the routing process during execution time preventing obstruction of the workflow. This will make the workflow continuous and unobstructed even in the absence of a particular user at any stage. This helps the organization to utilize fully the available resources. By allowing users to provision, manage, and de-provision their privileges unify the management of users, activities, and other resources. Such multi-domain user to dynamic user delegation of authority is desired in any adaptive and dynamic workflow system.

While provisioning delegation of authority, it is required that it should have minimum errors and ensures uniformity with all user permissions besides making delegation a simple, risk-free activity. Recent work [17] tries to add delegation extension to XACML 3.0 to express the right to administrate XACML policies within XACML itself using Administration and Delegation Profile. The ability to delegate administrative rights in XACML is new as of XACML 3.0. As adding an delegation constraint to a policy or rule, one issuer can restrict the authority of another issuer which is important while using delegated XACML rules as shown in Figure 9. XACML v3.0 Administration and Delegation Profile supports that <*PolicyIssuer*> element can be defined within the delegation rule and this helps us to separate the administrative delegation policy

```
<Policy PolicyId="Policy2"
Version="1.0"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overr
  <PolicyIssuer>
    <Attribute
      IncludeInResult="false"
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Carol</Attribut
    </Attribute>
  </PolicyIssuer>
  <Target>
    <AnyOf>
    <AnyOf>
    <AnyOf>
    <AnyOf>
  </Target>
  <Rule RuleId="Rule2" Effect="Permit">
    <Target/>
  </Rule>
</Policy>
```

**Figure 9 Basic XACML Delegation Format**

created at runtime during delegation from the general access control policy rules. The delegation profile draft explains how to negotiate for the right to issue a policy, but they have not provided any rules for removing a policy. So we need to adopt a secure and flexible revocation model in WfMS, which gives a delegating user i.e. delegator power to revert the privileges from delegatee. Both delegation and revocation need to take account of time constraints, so our system must account for this provision. As delegation can cause a critical security threat to a workflow system, provision and mitigation approaches need to be implemented on any WfMS. The key issue is evident in the real world scenario such as how to model the DOA in which one user can transfer his/ her authority to another user for a given period or a particular resource and then revoke it back.

# 5.Project Timeline

Oral presentation: May 2016
Project Report: October 2016
Final oral defense: December 2016

| **Date** | **Activity** | **Status** |
|---|---|---|
| August 2015 | Requirement Analysis and specification refinement | Done |
| September 2015 | Create and Design backend database using MongoDB | Done |
| November 2016 | Create REST services using JAX-RS, Morphia and Java and jUnit test cases | Done |
| December 2016 | Designing the frontend and backend page layout | Done |
| January 2016 | Integrate the REST web services with the web application | Done |
| February 2016 | Implementing Balana framework to the system and testing | Done |
| March 2015 | Implementation of approvable workflow steps via user's signatures | Done |
| April 2016 | Implementing the XACML policy rules to each action | Done |
| May 2016 | Implement the model for DOA and Obligations and verify the working conditions | On-going |
| June 2016 | Design the Model to implementing the Delegation of Authority based on XACML 3.0 | On-planning |
| August 2016 | Proof of Concept Testing and valid Measurement of the prototype | On-planning |
| October 2016 | Documentation write-up, Final presentation, and thesis defense | On-planning |

# 6. Bibliography


[1] S. Lakkaraju and D. Xu, “Integrated Modeling and Analysis of Attribute Based Access Control Policies and Workflows in Healthcare,” *2014 Int. Conf. Trust. Syst. their Appl.*, pp. 36–43, 2014.

[2] L. Sainan, “Task-role-based access control model and its implementation,” *2010 2nd Int. Conf. Educ. Technol. Comput.*, pp. V3–293–V3–296, 2010.

[3] Y. Liu, K. Xu, and J. Song, “A task-attribute-based workflow access control model,” *Proc. - 2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput. GreenCom-iThings-CPSCom 2013*, pp. 1330–1334, 2013.

[4] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, “Flexible Support for Multiple Access Control Policies,” *ACM Trans. ...*, vol. 26, no. 2, pp. 214–260, 2001.

[5] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong, “Access control in collaborative systems,” *ACM Comput. Surv.*, vol. 37, no. 1, pp. 29–41, 2005.

[6] Y. Lu and L. Zhang, “Domain administration of task-role based access control for process collaboration environments,” *5th Int. Conf. Inf. Assur. Secur. IAS 2009*, vol. 1, no. 1, pp. 643–647, 2009.

[7] J. Zhang, J. Sun, N. Li, and C. Hu, “Based on Mul-weighCted Roles in Worklsow System,” pp. 3–8, 2005.

[8] S. Chaari, F. Biennier, C. Ben Amar, and J. Favrel, “An authorization and access control model for workflow,” *First Int. Symp. Control. Commun. Signal Process. 2004.*, pp. 141–148, 2004.

[9] L. Wang, “Research of TRBAC model and the application in library management,” pp. 1–3, 2010.

[10] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-Based Access Control Models,” *IEEE Comput.*, vol. 29, no. 2, pp. 38–47, 1995.

[11] V. C. Hu, K. Scarfone, and R. Kuhn, *NIST Special Publication 800-162 DRAFT - FINAL Guide to Attribute Based Access Control ( ABAC ) Definition and Considerations NIST Special Publication 800-162 DRAFT - FINAL Guide to Attribute Based Access Control ( ABAC ) Definition and Considerations*. 2013.

[12] R. Abassi, F. Jacquemard, M. Rusinowitch, and S. G. El Fatmi, “XML access control: From XACML to annotated schemas,” *2010 2nd Int. Conf. Commun. Networking, ComNet 2010*, no. October 2015, 2010.

[13] M. Lischka, “Dynamic obligation specification and negotiation,” *Proc. 2010 IEEE/IFIP Netw. Oper. Manag. Symp. NOMS 2010*, pp. 155–162, 2010.

[14] D. Chadwick, “Obligation Standardization,” *Europe*, pp. 1–11, 2009.

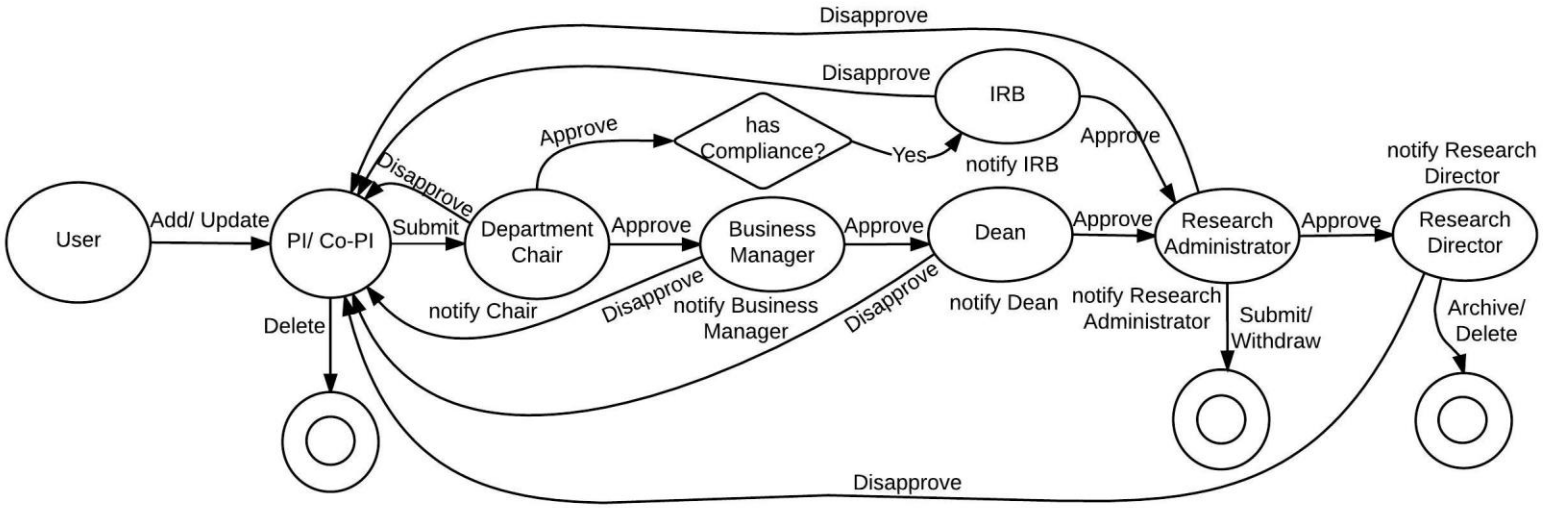
[15] P. H. Nguyen, G. Nain, J. Klein, T. Mouelhi, Y. Le Traon, and A. Weicker, “Model-Driven Adaptive Delegation ∗ Categories and Subject Descriptors,” pp. 61–72.

[16] G. Aucher, S. Barker, G. Boella, V. Genovese, and L. Van Der Torre, “Dynamics in delegation and revocation schemes: A logical approach,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6818 LNCS, pp. 90–105, 2011.

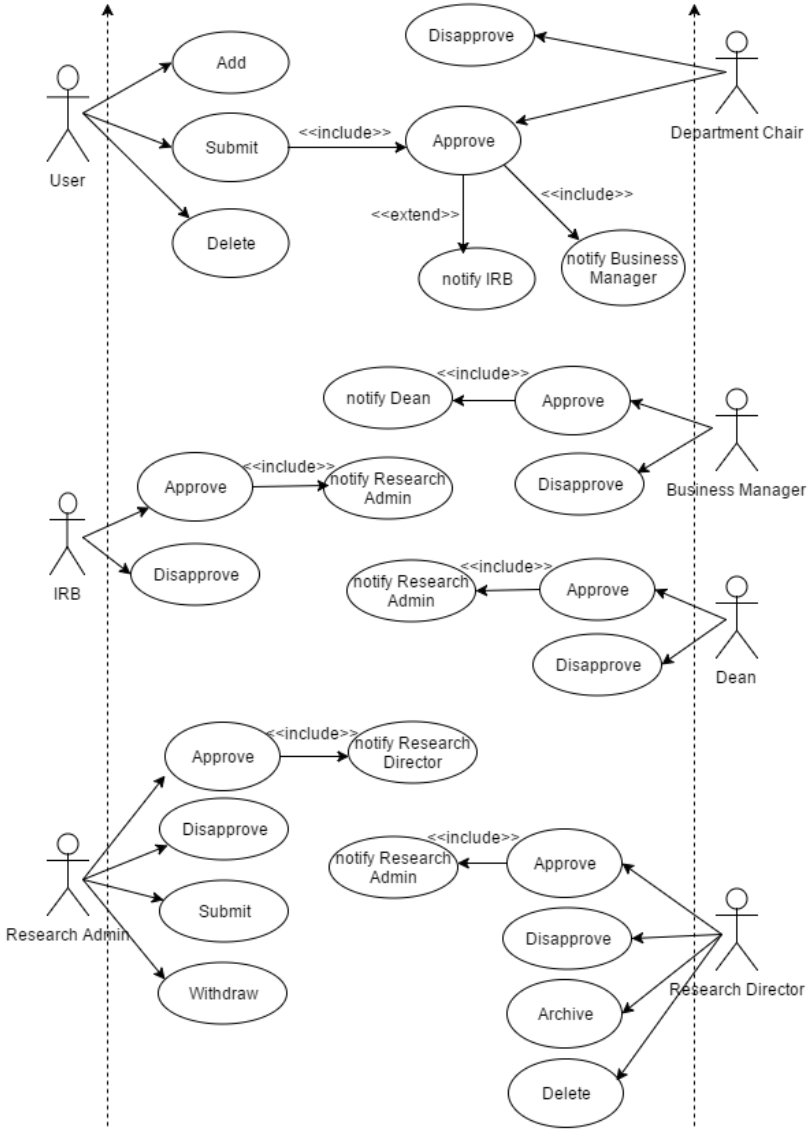
[17] Oasis, “eXtensible Access Control Markup Language,” *OASIS Stand.*, no. February, p. 141, 2005.

# 7. Appendix

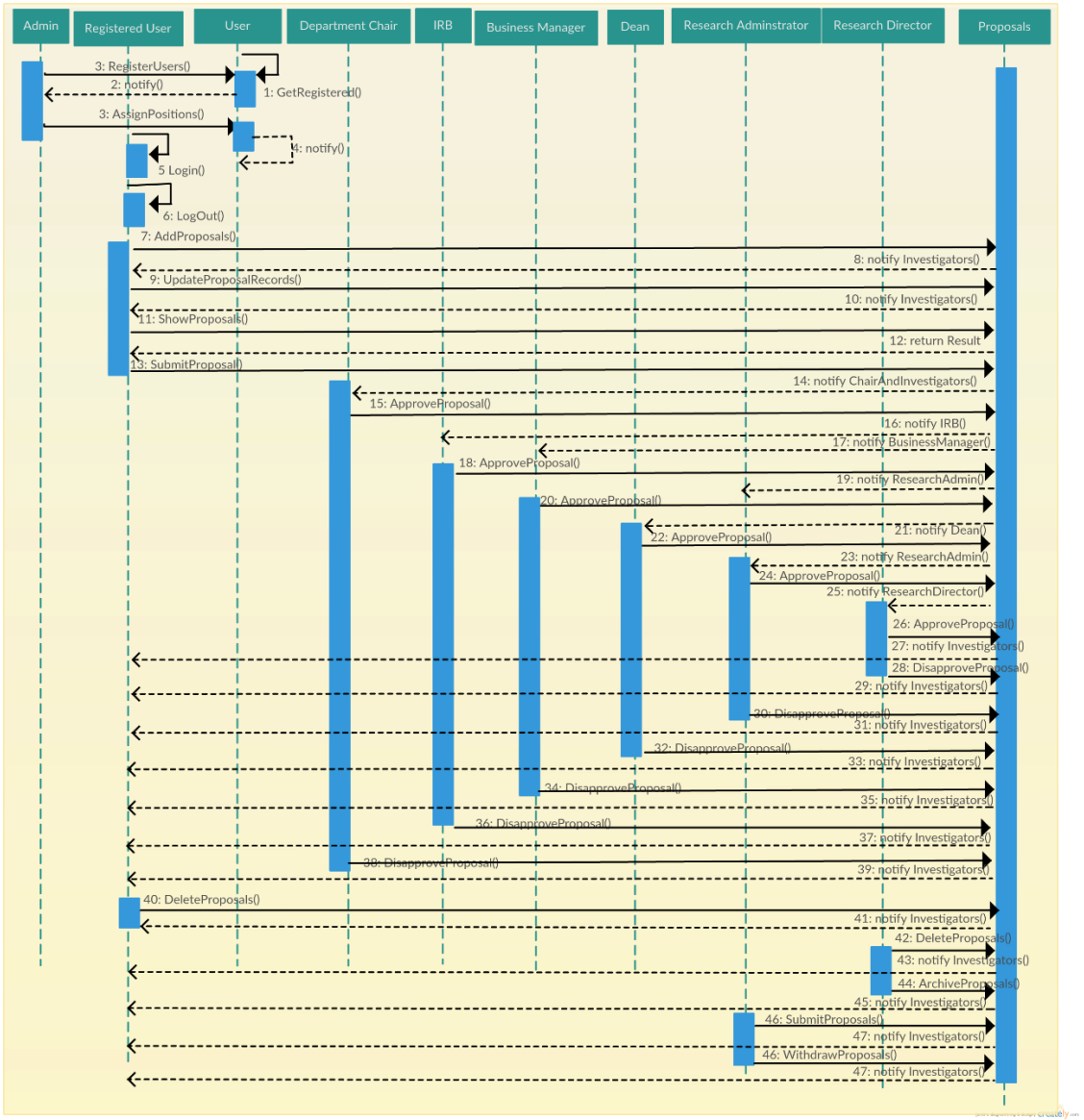
## I. State Diagram of GPWfMS



## II. Use Case Diagram of GPWfMS



## III. UML Sequence Diagram



## IV. Policy in XACML

a. "Department Chair" can "Approve" a "Whole Proposal" when ApprovedByDepartmentChair = READYFORAPPROVAL

With **PRE obligation**: Chair needs to Sign it first & **POST obligation**: Send Email to all Investigators such as PI, CO-PIs and Senior Personnel

# V. ABAC Policies with PRE and POST Obligations

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicyId="Proposal-Rules" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides" Version="1.0">
 <PolicyDefaults><XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</XPathVersion></PolicyDefaults>
 <Target />
 <Rule Effect="Permit" RuleId="ApproveProposalByDepartmentChair-Rule20">
  <Description> "Department Chair" can "Approve" a "Whole Proposal" when ApprovedByDepartmentChair= READYFORAPPROVAL with
                PRE obligation: Chair needs to Sign it first & POST obligation: Send Email to Business Manager </Description>
   <Target><AnyOf><AllOf> <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Department Chair</AtributeValue>
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:position.title" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false" />
      </Match>
      <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Whole Proposal</AttributeValue>
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:proposal.section" Category="urn:oasis:names:tc:xacml:1.0:attribute-category:re-
source" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false" />
      </Match>
      <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">READYFORAPPROVAL</AttributeValue>
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:ApprovedByDepartmentChair" Category="urn:oasis:names:tc:xacml:1.0:attribute-
category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false" />
      </Match>
      <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Approve</AttributeValue>
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:proposal.action" Category="urn:oasis:names:tc:xacml:1.0:attribute-category:action"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false" />
      </Match></AllOf></AnyOf></Target>
 </Rule>
 <ObligationExpressions>
   <ObligationExpression ObligationId="sendNotify" ObligationType="preobligation" FulfillOn="Permit">
     <AttributeAssignmentExpression AttributeId="obligationType">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">preobligation</AttributeValue>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="signedByCurrentUser">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:signedByCurrentUser/text()" DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
   </ObligationExpression>
   <ObligationExpression ObligationId="sendEmail" ObligationType="postobligation" FulfillOn="Permit">
     <AttributeAssignmentExpression AttributeId="obligationType">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">postobligation</AttributeValue>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="emailBody">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Hello User, your proposal has been updated.
</AttributeValue>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="emailSubject">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Notify updated successfully by:</AttributeValue>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="authorName">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:authorprofile/ak:fullname/text()" DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="piEmail">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:pi/ak:workemail/text()" DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="copisEmail">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:copis/ak:copi/ak:workemail/text()" DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="seniorsEmail">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:seniors/ak:senior/ak:workemail/text()" DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
   </ObligationExpression>
   <ObligationExpression FulfillOn="Deny" ObligationId="LogInvalidAccess">
     <AttributeAssignmentExpression AttributeId="invalidText">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Invalid access by: </AttributeValue>
     </AttributeAssignmentExpression>
     <AttributeAssignmentExpression AttributeId="authorName">
       <AttributeSelector MustBePresent="true" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="//ak:authorprofile/ak:fullname/text()"DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </AttributeAssignmentExpression>
   </ObligationExpression>
</ObligationExpressions>
</Policy>
```

## VI. Request with Obligations

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" CombinedDecision="false"
ReturnPolicyIdList="false">
  <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:proposal.role" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PI</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Content>
      <ak:record xmlns:ak="http://akpower.org">
        <ak:proposal>
          <ak:proposalid>5702a60865dbb30b09a492cf</ak:proposalid>
          <ak:signedByCurrentUser>true</ak:signedByCurrentUser>
          <ak:authorprofile>
            <ak:fullname>Milson Munakami</ak:fullname>
          </ak:authorprofile>
          <ak:pi>
            <ak:fullname>Milson Munakami</ak:fullname>
            <ak:workemail>milsonmun@yahoo.com</ak:workemail>
            <ak:userid>56fee3e965dbb35ce5c900fa</ak:userid>
          </ak:pi>
          <ak:copis>
            <ak:copi>…..</ak:copi>
          </ak:copis>
        </ak:proposal>
      </ak:record>
    </Content>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:3.0:multiple:content-selector" IncludeInResult="false">
      <AttributeValue XPathCategory="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
DataType="urn:oasis:names:tc:xacml:3.0:data-type:xpathExpression">//ak:record/ak:proposal</AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:DeletedByPI" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">NOTDELETED</AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:proposal.section" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Whole Proposal</AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:SubmittedByPI" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">NOTSUBMITTED</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:proposal.action" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Save</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

## VII. Response from XACML PEP with PRE and POST Obligations

```
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Result>
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
    <Obligations>
      <Obligation ObligationId="sendNotify">
        <AttributeAssignment  AttributeId="obligationType" DataType="http://www.w3.org/2001/XMLSchema#string">
          preobligation
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="signedByCurrentUser"
DataType="http://www.w3.org/2001/XMLSchema#string">
          true
        </AttributeAssignment>
      </Obligation>
      <Obligation ObligationId="sendNotify">
        <AttributeAssignment  AttributeId="obligationType" DataType="http://www.w3.org/2001/XMLSchema#string">
          preobligation
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="signedByCurrentUser"
DataType="http://www.w3.org/2001/XMLSchema#string">
          true
        </AttributeAssignment>
      </Obligation>
      <Obligation ObligationId="sendEmail">
        <AttributeAssignment  AttributeId="obligationType" DataType="http://www.w3.org/2001/XMLSchema#string">
          postobligation
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="emailBody" DataType="http://www.w3.org/2001/XMLSchema#string">
          Hello User,<br/><br/>Your proposal has been updated. As soon as possible please review your proposal
for any unwanted changes.<br/><br/>Thank you, <br/> GPMS Team
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="emailSubject" DataType="http://www.w3.org/2001/XMLSchema#string">
          Your proposal has been updated successfully by:
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="authorName" DataType="http://www.w3.org/2001/XMLSchema#string">
          Milson Munakami
        </AttributeAssignment>
        <AttributeAssignment  AttributeId="piEmail" DataType="http://www.w3.org/2001/XMLSchema#string">
          milsonmun@yahoo.com
        </AttributeAssignment>
        ….
      </Obligation>
    </Obligations>
  </Result>
</Response>
```

## VIII. OSP-Proposal-Data-Sheet

BOISE STATE UNIVERSITY

**Office of Sponsored Programs**
**Proposal Data Sheet**
Proposals must be submitted to OSP 3 working days prior to the proposal submission deadline.

**Proposal No.** (assigned by OSP) _____
**Date Received** (assigned by OSP) _____

**I. INVESTIGATOR INFORMATION**

| Role | Name | Position Title | Department/Center/Unit | College/Division | Phone # |
|---|---|---|---|---|---|
| PI | | | | Select | |
| Co-PI | | | | Select | |
| Co-PI | | | | Select | |
| Co-PI | | | | Select | |
| Co-PI | | | | Select | |

**II. PROJECT INFORMATION**

Project Title: _____

Project Type: ☐ Research-Basic ☐ Research-Applied ☐ Research-Development ☐ Instruction ☐ Other Sponsored Activity
(Please choose only one project type.)
Type of Request: ☐ Pre-Proposal ☐ New Proposal ☐ Continuation ☐ Supplement Due Date: _____
Project Period: From: _____ To: _____ Location of Project: ☐ Off-campus ☐ On-campus

**III. SPONSOR AND BUDGET INFORMATION**

Name of Granting Agency: _____
Direct Costs: $_____ F&A Costs: $_____ Total Costs: $_____ F&A Rate: _____%

**IV. COST SHARE INFORMATION**

Is institutional committed cost share included in the proposal? ☐ Yes ☐ No If Yes, complete the OSP Cost Share Form
Is Third Party committed cost share included in the proposal? ☐ Yes ☐ No

**V. UNIVERSITY COMMITMENTS**

Will new or renovated space/facilities be required? ☐ Yes ☐ No
Will rental space be required? ☐ Yes ☐ No
Does this project require institutional commitments beyond the end date of the project? ☐ Yes ☐ No
If yes, please refer to the OSP Proposal Data Sheet Instructions for required documentation.

**VI. CONFLICT OF INTEREST AND COMMITMENT INFORMATION**

Is there a financial conflict of interest **related to this proposal**? ☐ Yes ☐ No
If yes, has the financial conflict been disclosed? ☐ Yes ☐ No If no, your disclosure must be updated.
Has there been a material change to your annual disclosure from? ☐ Yes ☐ No If yes, your disclosure must be updated.

**VII. COMPLIANCE INFORMATION**

Does this project involve the use of Human Subjects? Provide IRB # or indicate pending. ☐ Yes ☐ No IRB # _____
Does this project involve the use of Vertebrate Animals? Provide IACUC # or indicate ☐ Yes ☐ No IACUC # _____
Does this project involve Biosafety concerns? Provide IBC# or indicate pending. ☐ Yes ☐ No IBC# _____
Does this project have Environmental Health & Safety concerns? ☐ Yes ☐ No

**VIII. ADDITIONAL INFORMATION**

Do you anticipate payment(s) to foreign nationals or on behalf of foreign nationals? ☐ Yes ☐ No
Do you anticipate course release time? ☐ Yes ☐ No
Are the proposed activities related to Center for Advanced Energy Studies? ☐ Yes ☐ No

**IX. COLLABORATION INFORMATION**

Does this project involve non-funded collaborations? ☐ Yes ☐ No If Yes, please list collaborating institutions/organizations below.
Collaborators: _____

**X. PROPRIETARY/CONFIDENTIAL INFORMATION**

Does this proposal contain any confidential information which is: ☐ Proprietary that should not be publicly released? ☐ No
☐ Yes, on pages _____ ☐ Patentable ☐ Copyrightable.
Will this project involve intellectual property in which the University may own or have an interest? ☐ Yes ☐ No
Note: Contact the Office of Technology Transfer for additional assistance on proprietary and patentable information at 208-426-5765.

## XI. CERTIFICATION/SIGNATURES

**Investigators, department chairs, directors, deans** certify that 1) the proposed activities are appropriate to the research, instruction, and public service mission of the University; 2) if funded all necessary resources as proposed will be provided for the project (i.e., cost share, personnel, facilities), and project expenditures that exceed the sponsor's award and/or payment upon completion of the project will be charged to the departmental account that you will identify at the time of award setup.

**Principal or Co-Principal Investigators** certify that 1) the information submitted within the application is true, complete and accurate to the best of the Investigator's knowledge; 2) all necessary resources to successfully complete the proposed project have been identified in the proposal; 3) the application is true, complete, and accurate to the best of my knowledge, 4) any false, fictitious, or fraudulent statements or claims may subject the PI to criminal, civil, or administrative penalties, 5) the PI agrees to accept responsibility for the scientific and programmatic conduct and financial oversight of the project and to provide the required progress reports; and 6) the PI shall use all reasonable and best efforts to comply with the terms, conditions, and policies of both the sponsor and the University. PIs should refer to http://web1.boisestate.edu/research/osp/standard-compliance.shtml for a list of responsibilities.

**Department chairs and deans** acknowledge that Facilities & Administrative costs for projects involving more than one college will be distributed in accordance with University policy 6100 unless otherwise directed in writing with approval from all deans involved.

| Principal/Co-Investigator(s) | Date | Dept Chair(s) or Director(s) | Date |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | Dean(s) | Date |
| | | | |
| | | | |

Business Manager (if applicable) has reviewed this proposal. Initials: _____

**Office of Sponsored Programs Administrative Use Only:**

Flow-Through, List Agency: _____

Funding Source: ☐ Federal ☐ Federal Flow-Through ☐ State of Idaho Entity ☐ Private for Profit
☐ Non-Profit Organization ☐ Non-Idaho State Entity ☐ College/University ☐ Local Entity
☐ Non-Idaho Local Entity ☐ Tribal Government ☐ Foreign

CFDA No.: _____ Program No: _____
Program/Solicitation Title: _____

Recovery: ☐ Full Recovery ☐ No Recovery-Normal Sponsor Policy ☐ No Recovery-Institutional Waiver
☐ Limited Recovery-Normal Sponsor Policy ☐ Limited Recovery-Institutional Waiver
Base: ☐ MTDC ☐ TDC ☐ TC ☐ Other ☐ N/A

Is PI salary included in the proposal? ☐ Yes ☐ No If No, provide a Department ID for 1% minimum
PI Salary: _____ PI Fringe: _____ Department ID: _____
Institutional Cost Share Documented ☐ Yes ☐ No ☐ NA Third Party Cost Share Documented ☐ Yes ☐ No ☐ NA

Are subrecipients (subcontracts/subawards) anticipated? ☐ Yes ☐ No
Names of subrecipients: _____

PI Eligibility Waiver on File ☐ Yes ☐ No ☐ NA ☐ This Proposal Only ☐ Blanket
Conflict of Interest Forms on File ☐ Yes ☐ No ☐ NA Excluded party list has been checked ☐ Yes ☐ No ☐ NA

Proposal Notes: Research Administrator: ☐ DF ☐ LG ☐ LN

Send original to Office of Sponsored Programs, MS 1135 or osp@boisestate.edu.
Please send email to osp@boisestate.edu. to request a final copy of the Proposal Data Sheet.