I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy
Enhanced Cloud Computing

Access Control Policy Generation From User Stories Using Machine Learning

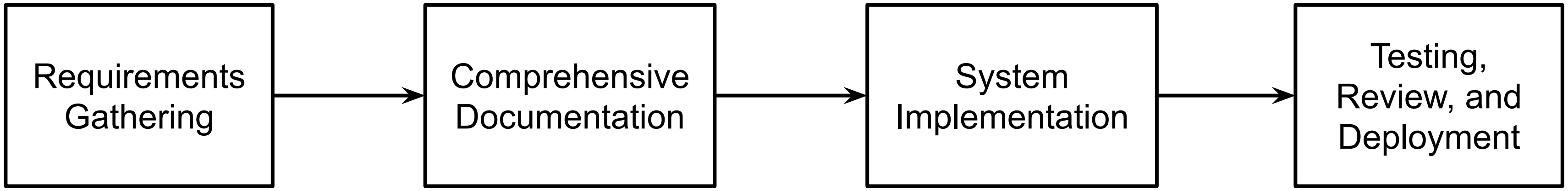
John Heaps[1], Ram Krishnan[2], Yufei Huang[3], Jianwei Niu[1], and Ravi Sandhu[1]

[1]Institute for Cyber Security (ICS), NSF Center for Security and Privacy Enhanced Cloud Computing (C-SPECC), and Department of Computer Science
[2]ICS, C-SPECC, and Department of Electrical and Computer Engineering
[3]ICS and Department of Electrical and Computer Engineering
The University of Texas at San Antonio

UTSA
Computer Science

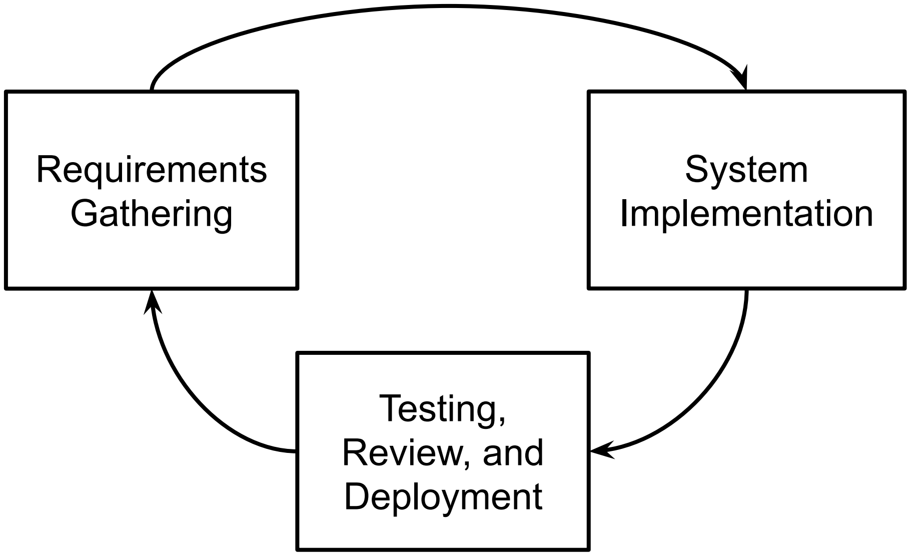
Agile Development

Traditional Software Development Methods



Agile Development

Agile Software Development Methods



Security Concerns of Agile Development

- Agile development propagates vulnerability issues
  - Constant changes in requirements
  - Frequent code refactoring
  - Lack of documentation
  - Speed of development

- How to help stakeholders during development to overcome the propagation of vulnerabilities?
  - Previous literature has suggested the manual creation of additional documentation
  - Our approach is to automatically generate additional documentation

User Stories

- Used to define the requirements of a system from the actor (or user) perspective

- Simple
  - As a system admin, I want to create a new user account.

- No Access Control
  - As an Older Person, I want to use only well-visible buttons.

- Multi-Functionality
  - As a camp administrator, I want to be able to see all my camp groups and the events scheduled for each camp group, so that I can notify counselors of what their group will be doing for the day.

Using User Stories for Documentation Generation

- User stories are the only artifacts required by agile development
- We focus on access control policy in our initial research
- **What access control information do user stories contain, and how can that information be identified, extracted, and presented to stakeholders?**
    - We will use deep learning to identify and extract access control information from user stories

Dataset

- Dalpiaz[1] Dataset
  - 1600 user stories
  - 14 different projects (50 - 130 user stories per project)
  - Project diversity
    - Elderly care
    - Data management platform
    - Administrative management

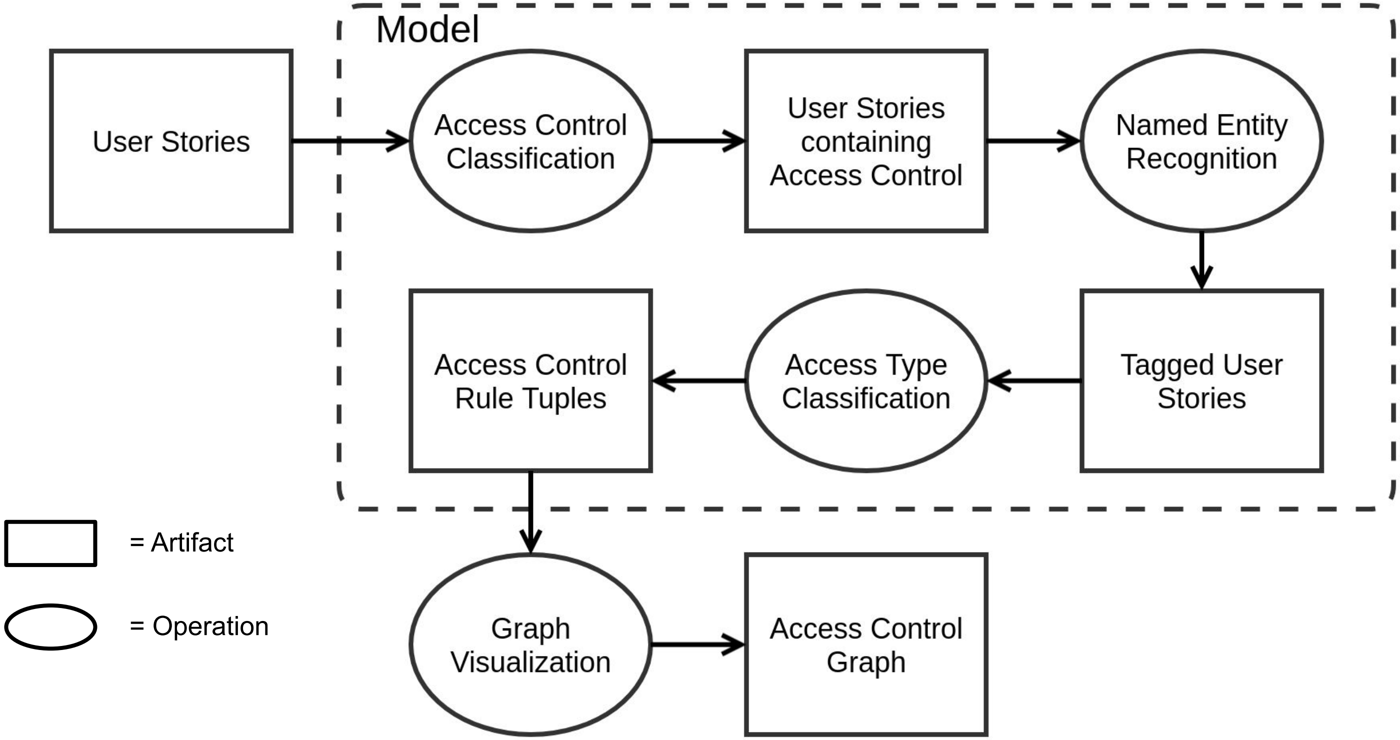
[1] Dalpiaz, F., Van der Schalk, I., Lucassen, G.: Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and nlp. In:International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 119–135. Springer (2018)

User Story to Access Control Tuple

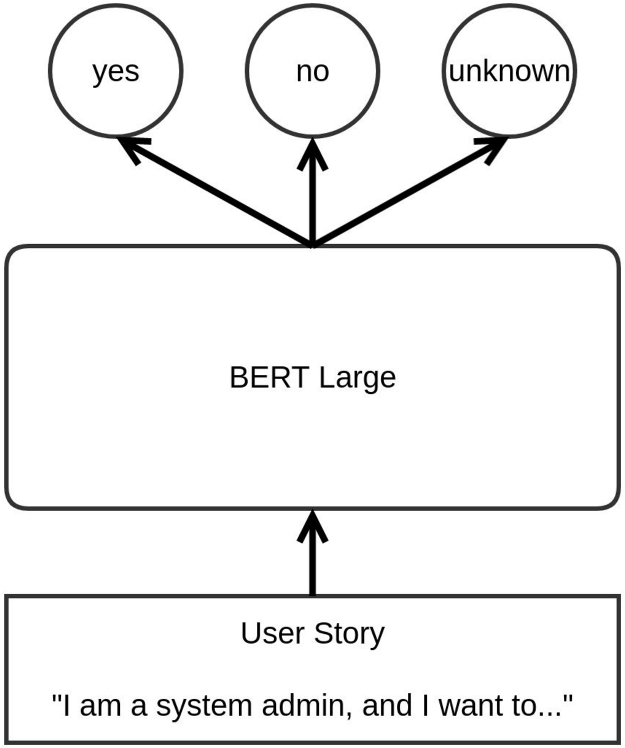
- Input is a user story
    - As a camp administrator, I want to be able to create, modify rules that campers and camp workers have to follow.

- Final output is a set of tuples that represent the access control in the user story as (Actor, Data Object or Operation, Type of Access)
    - (Camp Administrator, Rules (Data Object), create edit view)
    - (Camper, Rules (Data Object), view)
    - (Camp Worker, Rules (Data Object), view)

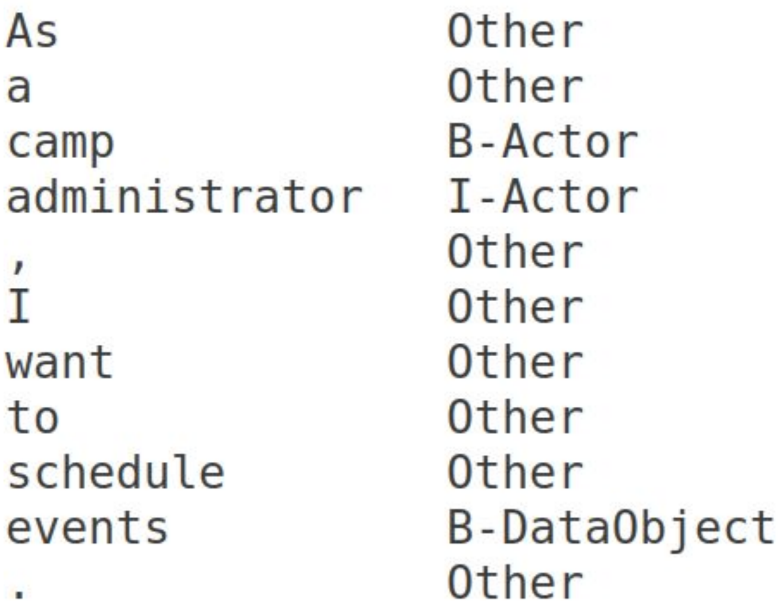
Approach


Model
User Stories
Access Control Classification
User Stories containing Access Control
Named Entity Recognition
Tagged User Stories
Access Type Classification
Access Control Rule Tuples
Graph Visualization
Access Control Graph
= Artifact
= Operation

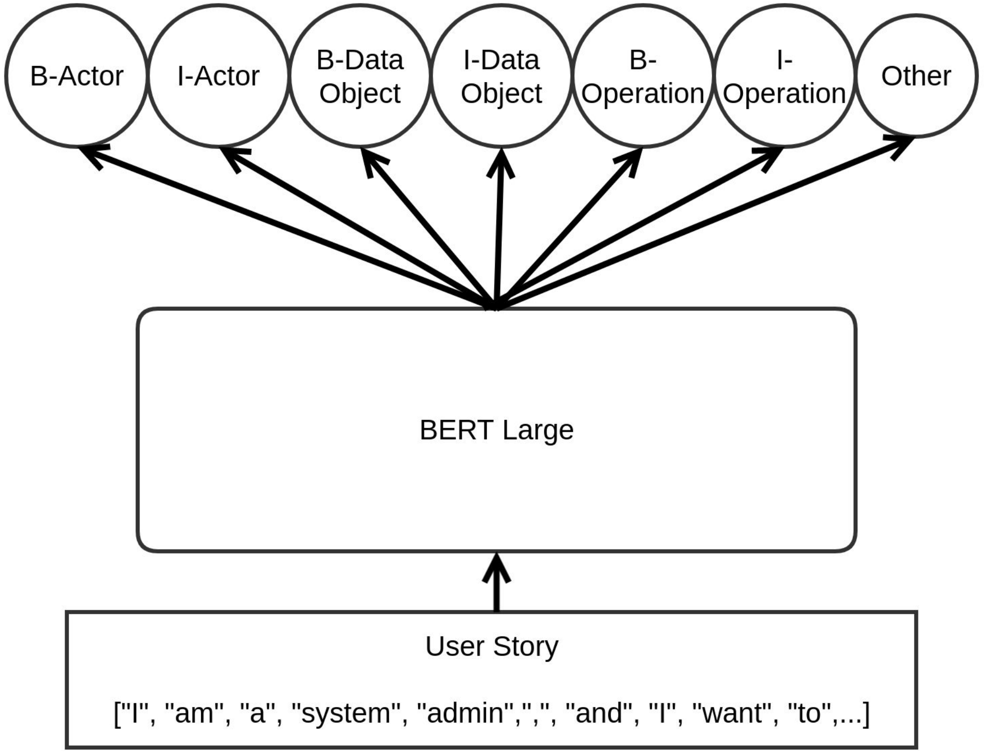
Component 1 - Access Control Classification



Component 2 - Named Entity Recognition



As	Other
a	Other
camp	B-Actor
administrator	I-Actor
,	Other
I	Other
want	Other
to	Other
schedule	Other
events	B-DataObject
.	Other



B-Actor

I-Actor

B-Data Object

I-Data Object

B-Operation

I-Operation

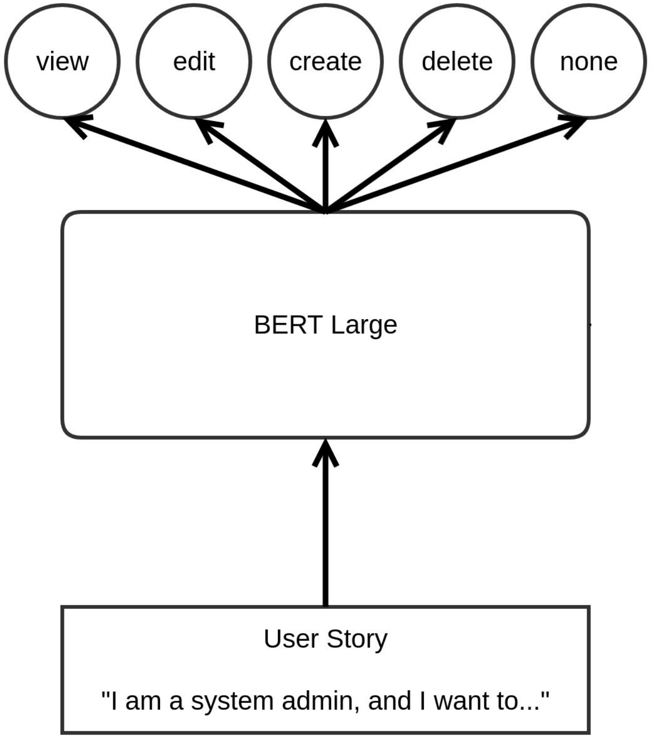
Other

BERT Large

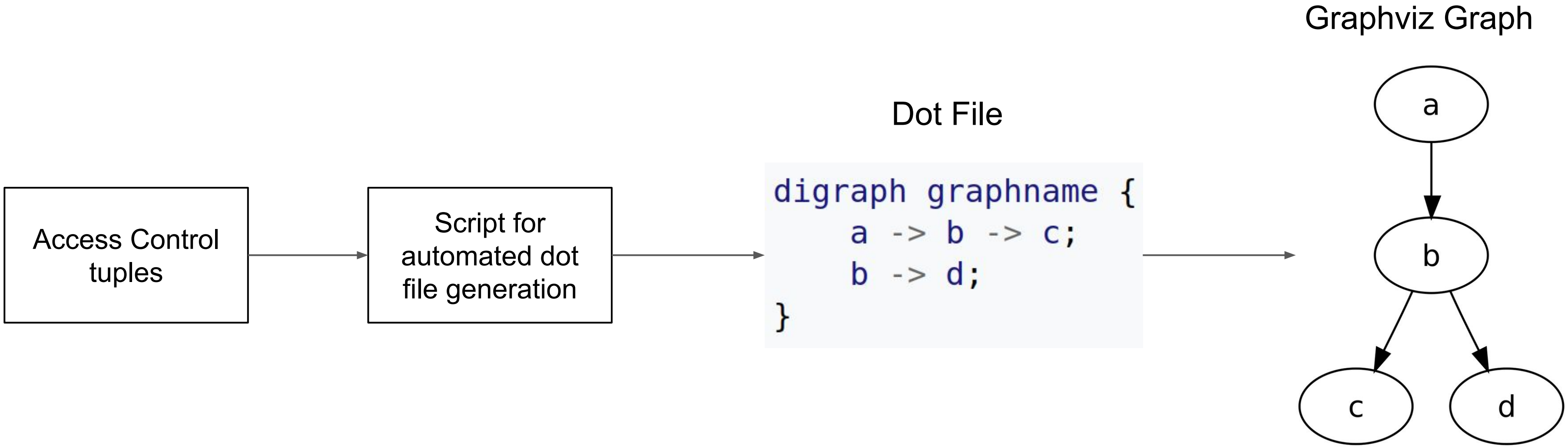
User Story

["I", "am", "a", "system", "admin",",", "and", "I", "want", "to",...]

Component 3 - Access Type Classification



Visualization



Access Control tuples

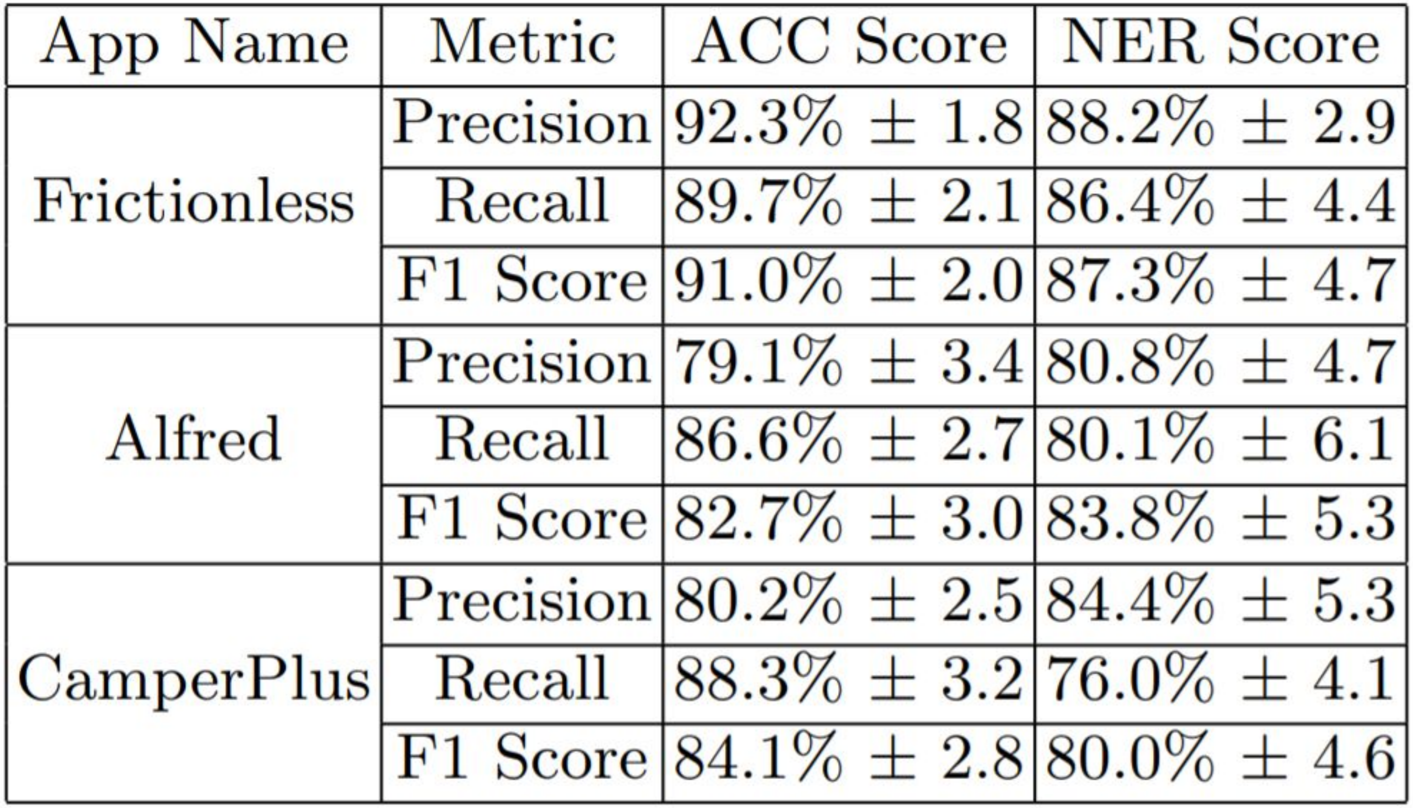
Script for automated dot file generation

Dot File

```
digraph graphname {
     a -> b -> c;
     b -> d;
}
```

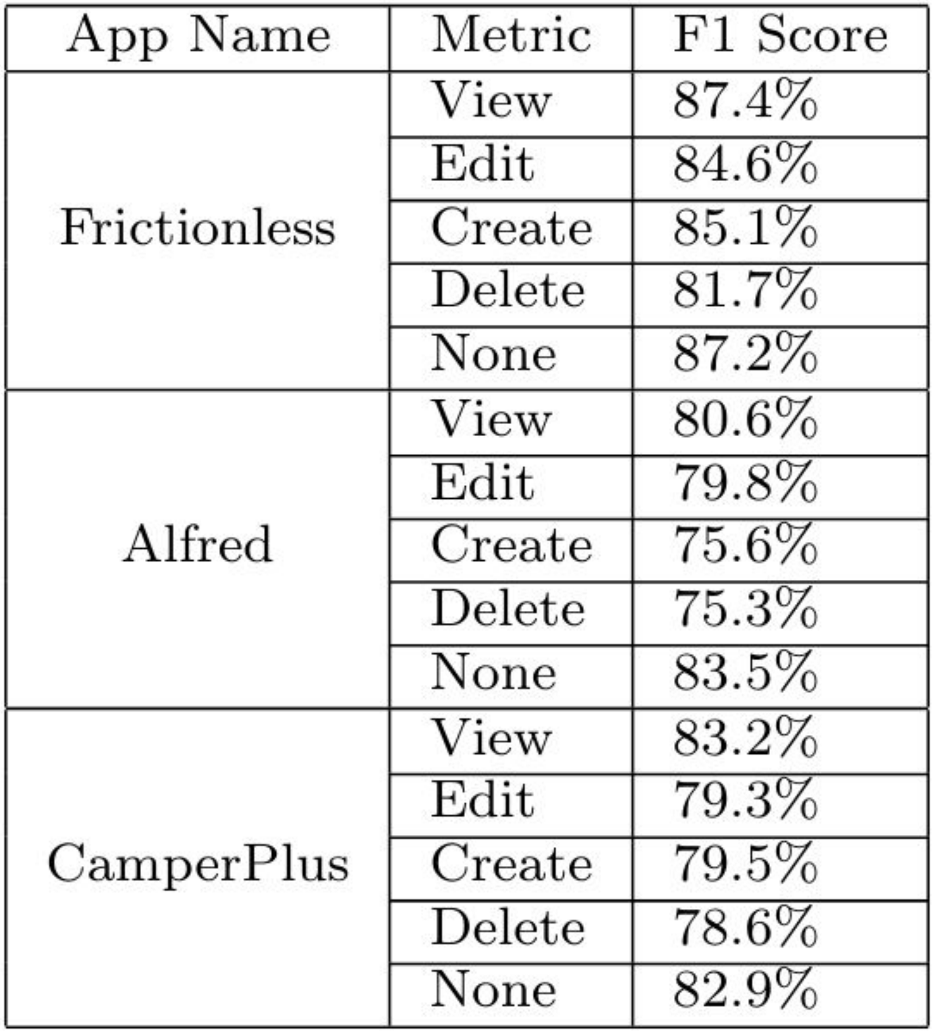
Graphviz Graph

Results - Access Control Classification and Named Entity Recognition



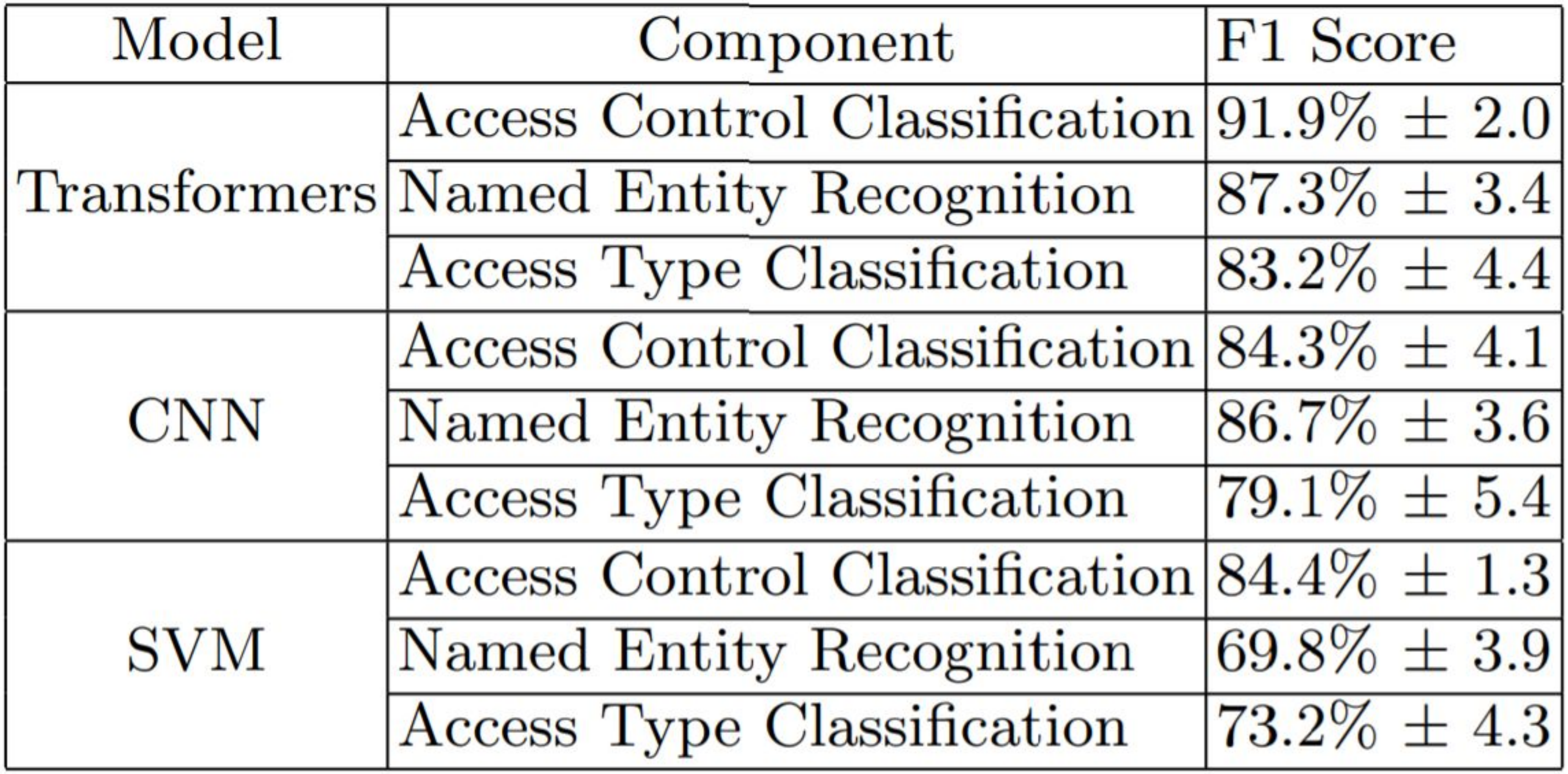
App Name	Metric	ACC Score	NER Score
Frictionless	Precision	92.3% ± 1.8	88.2% ± 2.9
	Recall	89.7% ± 2.1	86.4% ± 4.4
	F1 Score	91.0% ± 2.0	87.3% ± 4.7
Alfred	Precision	79.1% ± 3.4	80.8% ± 4.7
	Recall	86.6% ± 2.7	80.1% ± 6.1
	F1 Score	82.7% ± 3.0	83.8% ± 5.3
CamperPlus	Precision	80.2% ± 2.5	84.4% ± 5.3
	Recall	88.3% ± 3.2	76.0% ± 4.1
	F1 Score	84.1% ± 2.8	80.0% ± 4.6

Results - Access Type Classification



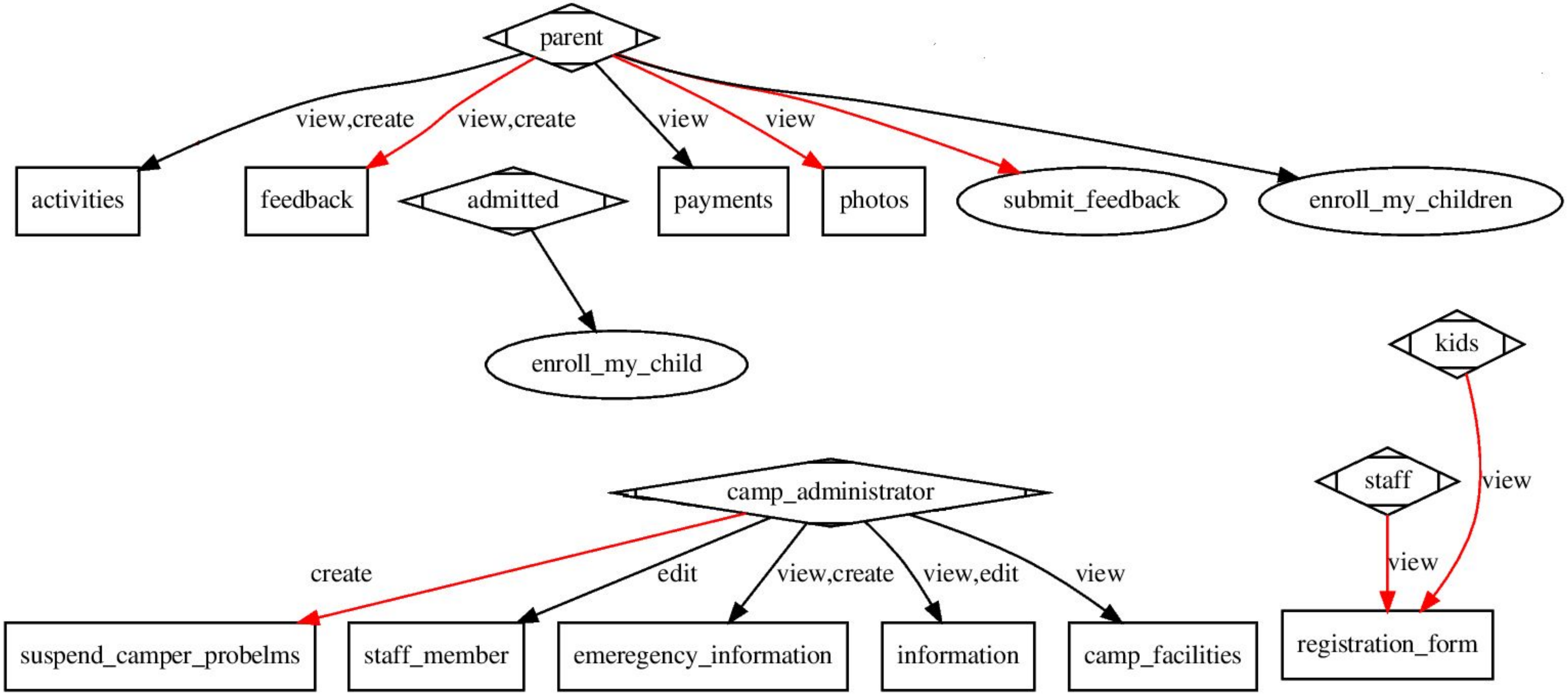
App Name	Metric	F1 Score
Frictionless	View	87.4%
	Edit	84.6%
	Create	85.1%
	Delete	81.7%
	None	87.2%
Alfred	View	80.6%
	Edit	79.8%
	Create	75.6%
	Delete	75.3%
	None	83.5%
CamperPlus	View	83.2%
	Edit	79.3%
	Create	79.5%
	Delete	78.6%
	None	82.9%

Results - Model Comparison



Model	Component	F1 Score
Transformers	Access Control Classification	91.9% ± 2.0
	Named Entity Recognition	87.3% ± 3.4
	Access Type Classification	83.2% ± 4.4
CNN	Access Control Classification	84.3% ± 4.1
	Named Entity Recognition	86.7% ± 3.6
	Access Type Classification	79.1% ± 5.4
SVM	Access Control Classification	84.4% ± 1.3
	Named Entity Recognition	69.8% ± 3.9
	Access Type Classification	73.2% ± 4.3

Results - Visualization



Conclusion and Future Work

- We have shown that access control information and policy can be identified and extracted from user stories

- Future Work
    - Showing changes in access control throughout the agile process
    - Human interactivity
    - Active Learning
    - Other types of documentation generation