Secure and Auditable Academic Collections Storage via Hyperledger Fabric-Based Smart Contracts

Thomas Atkins

B.S in Computer Engineering

Submitted to the graduate degree program in Department of Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master Of Computer Engineering.

Drew Davidson, Chairperson

Committee members

Bo Luo

Fengjun Li

Date defended: December 11, 2023

The Thesis Committee for Thomas Atkins certifies
that this is the approved version of the following thesis:

Secure and Auditable Academic Collections Storage via Hyperledger Fabric-Based Smart Contracts

Drew Davidson, Chairperson

Date approved: _____

Abstract

This paper introduces a novel approach to manage collections of artifacts through smart contract access control, rooted in on-chain role-based property-level access control. This smart contract facilitates the lifecycle of these artifacts including allowing for the creation, modification, removal, and historical auditing of the artifacts through both direct and suggested actions. This method introduces a collection object designed to store role privileges concerning state object properties. User roles are defined within an on-chain entity that maps users' signed identities to roles across different collections, enabling a single user to assume varying roles in distinct collections. Unlike existing key-level endorsement mechanisms, this approach offers finer-grained privileges by defining them on a per-property basis, not at the key level. The outcome is a more flexible and fine-grained access control system seamlessly integrated into the smart contract itself, empowering administrators to manage access with precision and adaptability across diverse organizational contexts. This has the added benefit of allowing for the auditing of not only the history of the artifacts, but also for the permissions granted to the users.

Acknowledgements

Acknowledgements go here.

Contents

List of Figures

List of Tables

Chapter 1

Chapter 1 Title

Abstract

Use the chapterabstract environment, not the abstract environment, if you want to plant an abstract at the top of the chapter.

1.1 Introduction

If needed Here's a quote environment.

Here's a citation, so we don't get a "no citation warning" **?**. Here's a figure. Here how to reference a figure such as Figure **??**.

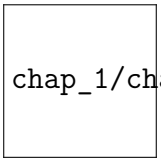
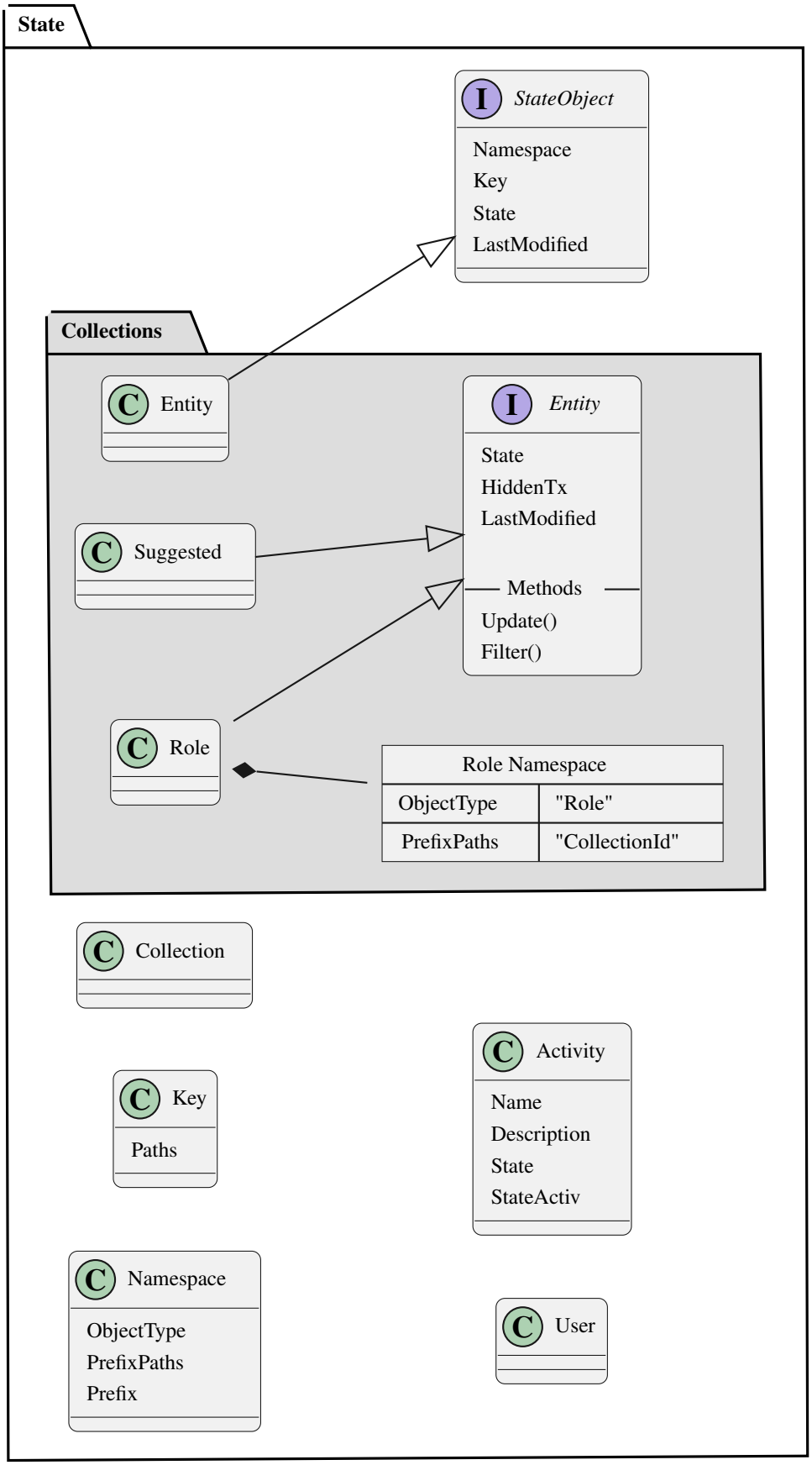


Figure 1.1: Figure for list of figures in the content page.


State
StateObject
Namespace
Key
State
LastModified
Collections
Entity
Entity
State
HiddenTx
LastModified
Suggested
Methods
Update()
Filter()
Role
Role Namespace
ObjectType
"Role"
PrefixPaths
"CollectionId"
Collection
Activity
Name
Description
State
StateActiv
Key
Paths
Namespace
ObjectType
PrefixPaths
Prefix
User

	R: polr	R: lrm	SAS	Stata
$\hat{b}_1$	-0.28	-0.28	0.28	-0.28
$\hat{\zeta}_1$	-4.24	4.24	-4.24	-4.24
$\hat{\zeta}_2$	-2.32	2.32	-2.32	-2.32

Table 1.1: Table for the list of tables in the contents page.

Type Of Object	B1	C1
Collections	B2	C2
A3	B3	C3

Table 1.2: Collections Items Keys

Here's a table.

Chapter 2

Chapter 2 Title

Abstract

Chapter 2 Abstract: Code and figure example.

2.1 Example Section

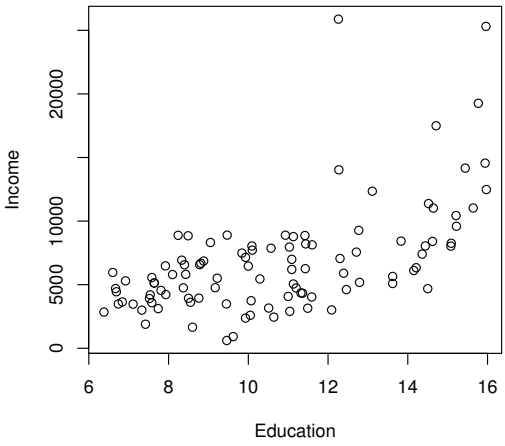
2.1.1 Code Example

Check on conventions on inputting code just to be safe. This is likely field dependent, so this is worth considering. This is the default style, which is aberrant to look at. In any case, this is how subimport works for nested files to organize your document with each chapter self contained in it's own folder.

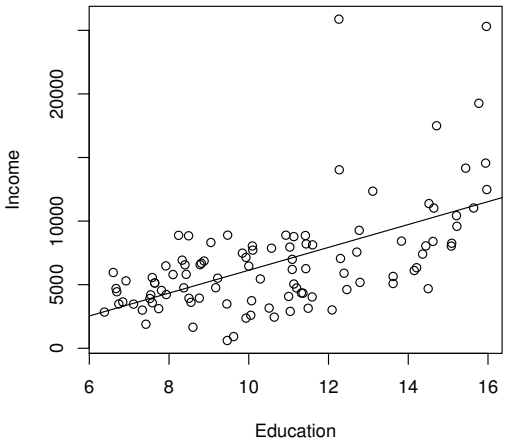
```
library(car)


pdf(file="car.inc.ed.pdf", height=5,width=5, onefile=F,
    paper="special")
plot(income~education, xlab="Education", ylab="Income",
    main="", data=Prestige)
dev.off()
```

2.1.2 Subfigure Example.

This is a true subfigure example.



(a) Caption A



(b) Caption B

Chapter 3

Conclusions

Abstract

Chapter 2 Abstract: Code and figure example.

References

Appendix A

Misc stuff

Appendix B

Misc Stuff 2