CMSC388N:

Build It, Break It, Fix It: Competing to Secure Software

Lecture 1

Prof. Daniel Votipka

Winter 2020

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

The Plan

- Introductions
- Course Overview and Logistics
- Project Description
- Threat Modeling

Why is Secure Development important?

Why does it matter?

- Enterprise Security
  - National Security
    - Financial Sector
    - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Why does it matter?

- Enterprise Security
  - National Security
  - Financial Sector
    - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Why does it matter?

- Enterprise Security
  - National Security
  - Financial Sector
    - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Why does it matter?

- Enterprise Security
  - National Security
  - Financial Sector
  - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Why does it matter?

- Enterprise Security
  - National Security
  - Financial Sector
  - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Some bad news

Designing secure systems is difficult

Designing secure systems is difficult.

Fundamental asymmetry between attacker and defender

Functionality is easy to measure, but…

Airplane works

Airplane doesn't work

…*security* is hard to measure

Web browser 0wned

BIBIFI
CMSC 388N: Overview

Home Schedule Projects Resources Syllabus Piazza

Build it, Break it, Fix it: Competing to Secure Software - Winter 2020

Instructors	Daniel Votipka, Kelsey Fulton, Mike Hicks, and Michelle Mazurek
Time	M/W/F 12:00-1:30pm
Location	CSIC 3120
Office Hours	W 1:45-2:45pm, IRB 5112

Course Description

This will mostly be a hands-on course centered on the design, implementation, and refinement of a single medium-sized system of programs throughout the semester. The course will be structured into two phases. In the first part of the course, you will spend most of your time in class designing and building your secure system. Then, in the second part, you will attempt to break your classmates' programs and respond to exploits by classmates (by fixing your code).

Throughout, we plan to cover a subset of the secure software development material typically covered in 414 (e.g., network security, cryptography, memory corruption attacks, etc.). However, most of the class time will be spent working on the project in teams. Our hope with this flipped classroom approach is that we'll be able to provide tailored, in-depth feedback as we go through to help everyone get a solid, practical understanding of software security by the end of the course.

Additionally, data generated throughout the course of the class will be used in ongoing secure development learning research. This includes recordings and observations of team discussions and any submitted artifacts (code changes, design documents, etc.). All data collected will be anonymized prior to reporting and raw data will be maintained in a secure location only accessible to the instructors and research team. You will not be required to perform any tasks beyond what would be required for an equivalent course; we will simply maintain and report on the anonymized data you produce.

Web browser not 0wned

BIBIFI
CMSC 388N: Overview

Home Schedule Projects Resources Syllabus Piazza

Build it, Break it, Fix it: Competing to Secure Software - Winter 2020

Instructors	Daniel Votipka, Kelsey Fulton, Mike Hicks, and Michelle Mazurek
Time	M/W/F 12:00-1:30pm
Location	CSIC 3120
Office Hours	W 1:45-2:45pm, IRB 5112

Course Description

This will mostly be a hands-on course centered on the design, implementation, and refinement of a single medium-sized system of programs throughout the semester. The course will be structured into two phases. In the first part of the course, you will spend most of your time in class designing and building your secure system. Then, in the second part, you will attempt to break your classmates' programs and respond to exploits by classmates (by fixing your code).

Throughout, we plan to cover a subset of the secure software development material typically covered in 414 (e.g., network security, cryptography, memory corruption attacks, etc.). However, most of the class time will be spent working on the project in teams. Our hope with this flipped classroom approach is that we'll be able to provide tailored, in-depth feedback as we go through to help everyone get a solid, practical understanding of software security by the end of the course.

Additionally, data generated throughout the course of the class will be used in ongoing secure development learning research. This includes recordings and observations of team discussions and any submitted artifacts (code changes, design documents, etc.). All data collected will be anonymized prior to reporting and raw data will be maintained in a secure location only accessible to the instructors and research team. You will not be required to perform any tasks beyond what would be required for an equivalent course; we will simply maintain and report on the anonymized data you produce.

Some good news

Computer security is a growth area.

Awesome

Course Goals

- Learn how to design more robust systems
- Learn how to protect against attacks
- Think like the bad actor, behave like the good actor

This course provides hands-on practice developing and exploiting secure systems. Students will be asked to develop a secure IoT system and build a better understanding of secure design and implementation through doing.

Course Goals

- Learn how to design more robust systems
- Learn how to protect against attacks
- Think like the bad actor, behave like the good actor

This course provides hands-on practice developing and exploiting secure systems. Students will be asked to develop a secure IoT system and build a better understanding of secure design and implementation through doing.

This is compressed semester, so **we expect a lot**, but we think it will be very beneficial and hopefully a little fun!

Non-goals

- Familiarization with latest tools
- Professional security certification

Course as Research

- This course is part of our research studying how students learn about and write secure programs
- All information collected during discussions and through assignments will be used in our research
- All data will be anonymized to prevent association of your identity to results

Instructor Team

Michelle Mazurek

Michael Hicks

Kelsey Fulton

Dan Votipka

Instructor Team

Michelle Mazurek

Michael Hicks

Kelsey Fulton

Dan Votipka

Ice Breaker!

Ice Breaker!

- Favorite programming language
- Least favorite programming language
- Favorite time to work

Ice Breaker!

Think about who you might want on your team!

- Favorite programming language
- Least favorite programming language
- Favorite time to work

Ice Breaker!

Think about who you might want on your team!

- Favorite programming language
- Least favorite programming language
- Favorite time to work

Ice Breaker!

Think about who you might want on your team!

- Favorite programming language
- Least favorite programming language
- Favorite time to work

Ice Breaker!

Think about who you might want on your team!

- Favorite programming language
- Least favorite programming language
- Favorite time to work

Course policies, expectations, and other fun bureaucratic goodness

Course website:

https://www.cs.umd.edu/class/winter2020/cmsc388N/

This is the most important slide in this deck!

Course website:

https://www.cs.umd.edu/class/winter2020/cmsc388N/

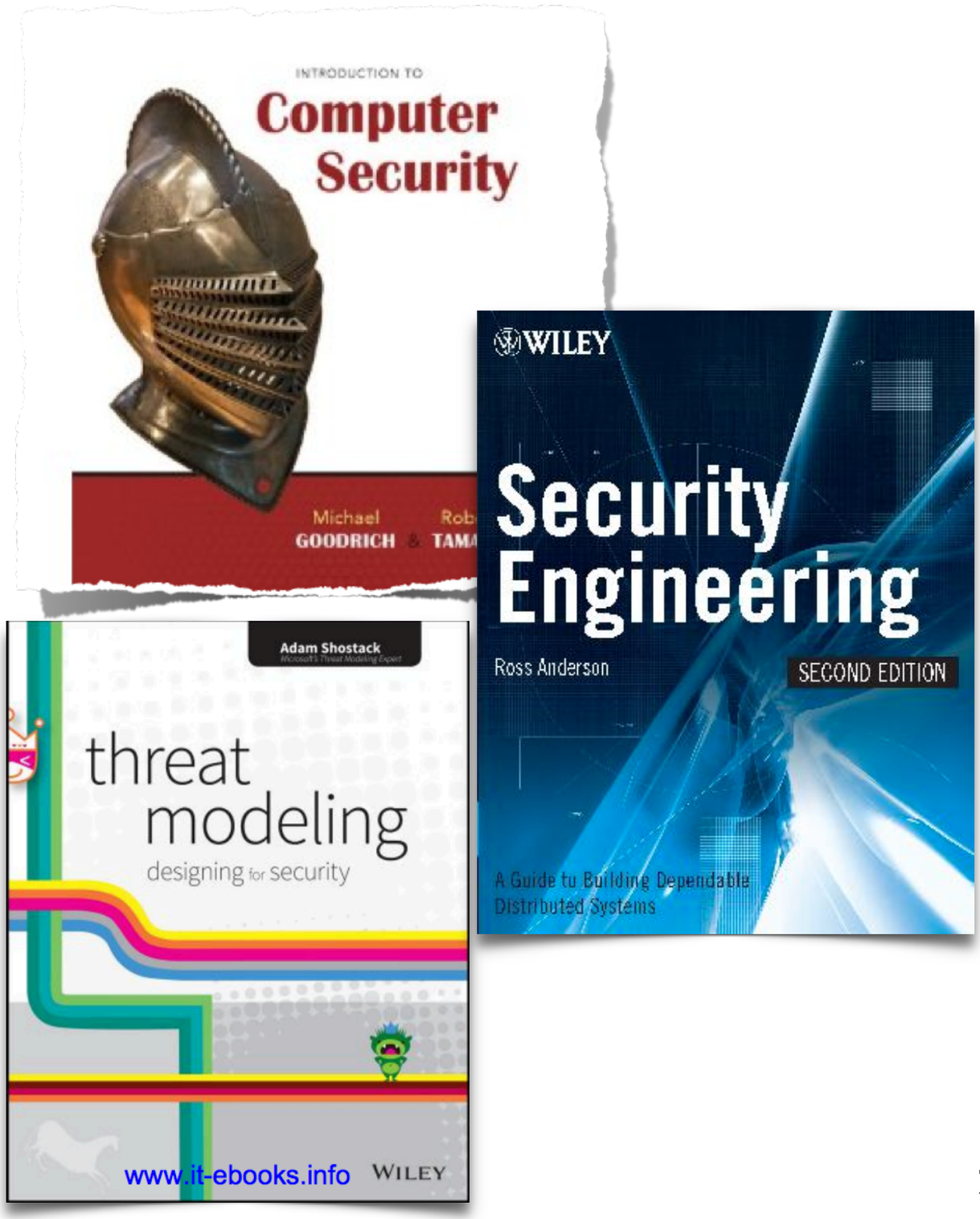
Prerequisites

- CMSC216 and CMSC250
- You will build stuff. I expect you to:
  - know how to code
  - be(come) comfortable with Linux/UNIX/git

Office Hours

- Available on request
- Location: IRB 5112

Textbook



- There is **no** required textbook for this course.
- Some helpful texts:
  - Introduction to Computer Security by Goodrich and Tamassia
  - Security Engineering by Ross Anderson (available online)
  - Threat Modeling: Designing for Security by Adam Shostack (available online)

Things that are not your textbook

WIKIPEDIA
The Free Encyclopedia

Slashdot
News for Nerds. Stuff that matters.

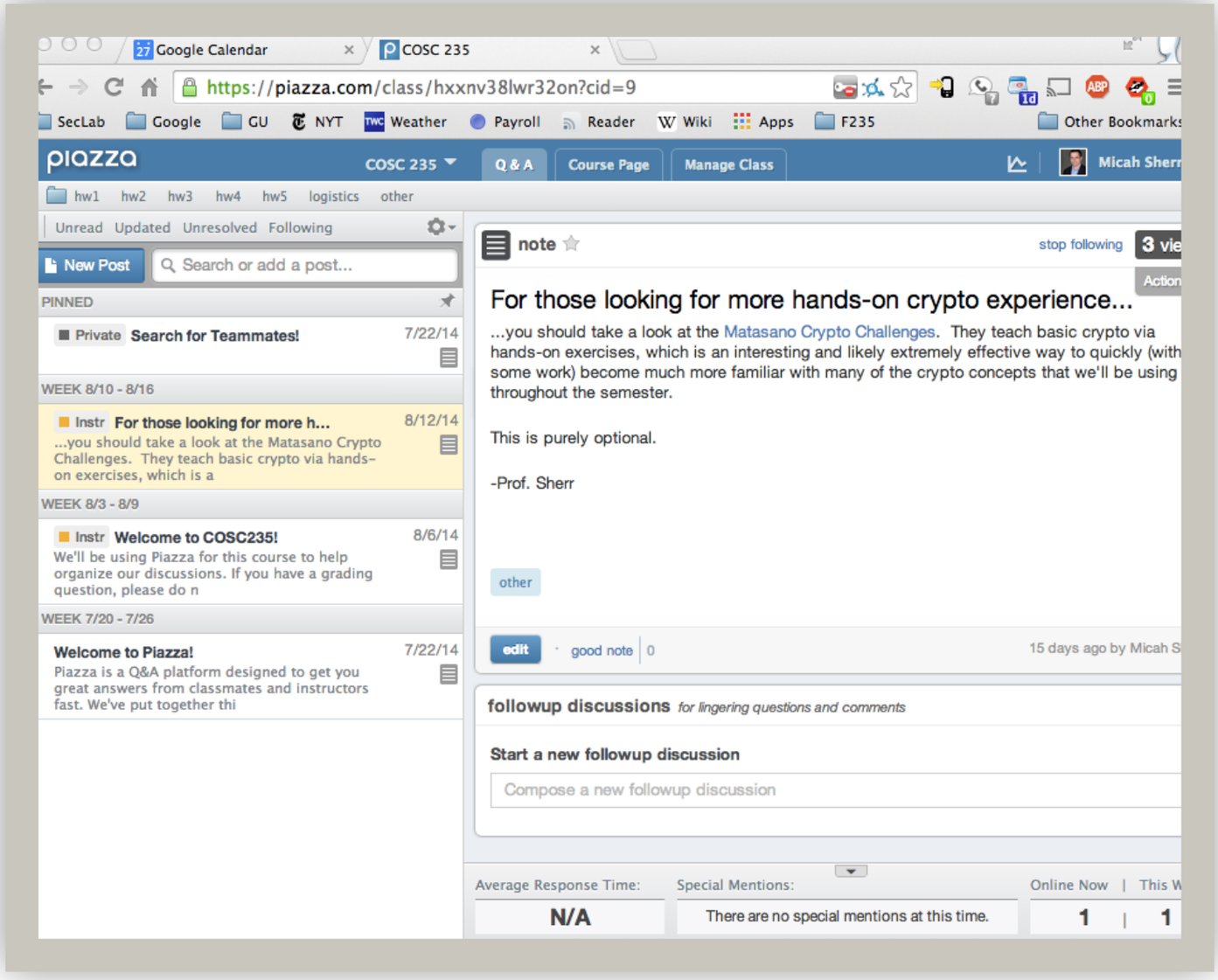
digg

Google

StackExchange

Online Course Discussion

- **Extensive** class discussions and announcements via Piazza
  - Be prepared to receive many emails
- You are expected to read each and every posting *(during the workday)*
- See course webpage for Piazza URL.



Online Course Discussion

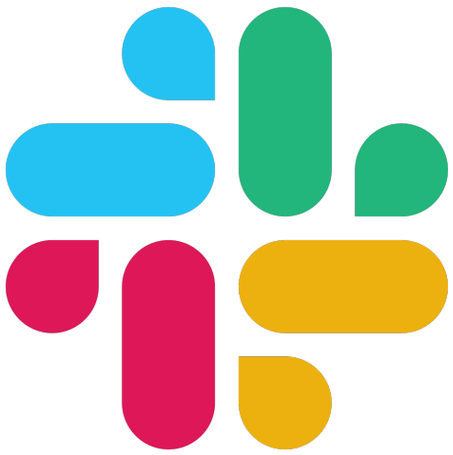
- Post to Piazza if…
  - …you have a question about the class subject matter (slides, lectures, etc.)
  - …you need a clarification on a homework or project specification
  - …you have a general question about secure development
  - …you have a question regarding a class policy
- If you send any of the above to me directly, I'll ask you to post it on Piazza
- *Don't*:
  - Give away project implementations details
  - Start flamewars
- *Do* be respectful of others

Emailing

- It's really best not to email me. Emails get lost. Piazza posts stay there until I actually resolve them.
- Send a <u>private</u> Piazza post if…
- …you have a grading issue
- …you need to ask a question that would reveal a partial/complete solution to a homework problem

Intra-Team Communication

- Slack will be the **required** channel for intra-team communication
- Each team will have a private channel
- Channels will be created for you once you select teams



UMD CMSC388N

umdcmsc388n.slack.com

Excused Absences

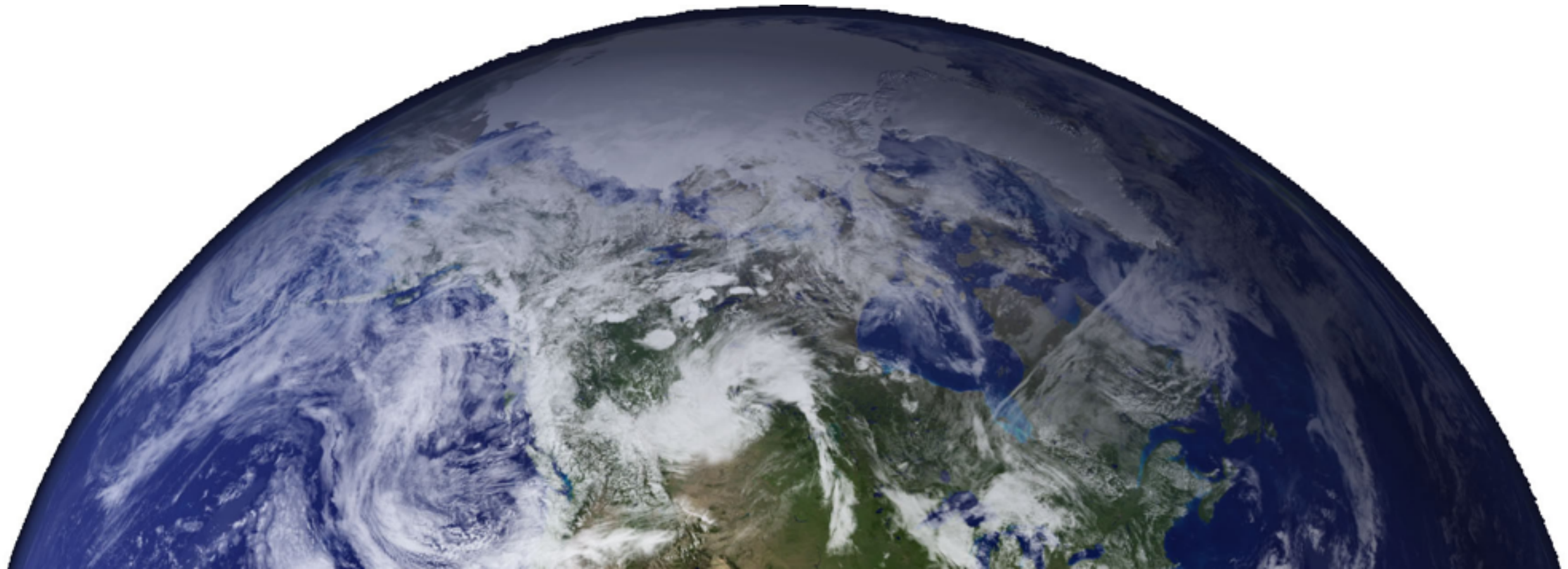
- Most class time will be dedicated to completing the semester long project, so attendance is essential to your success in this course.

- Reasons for excused absences: Religious observation, illness, personal or family emergency

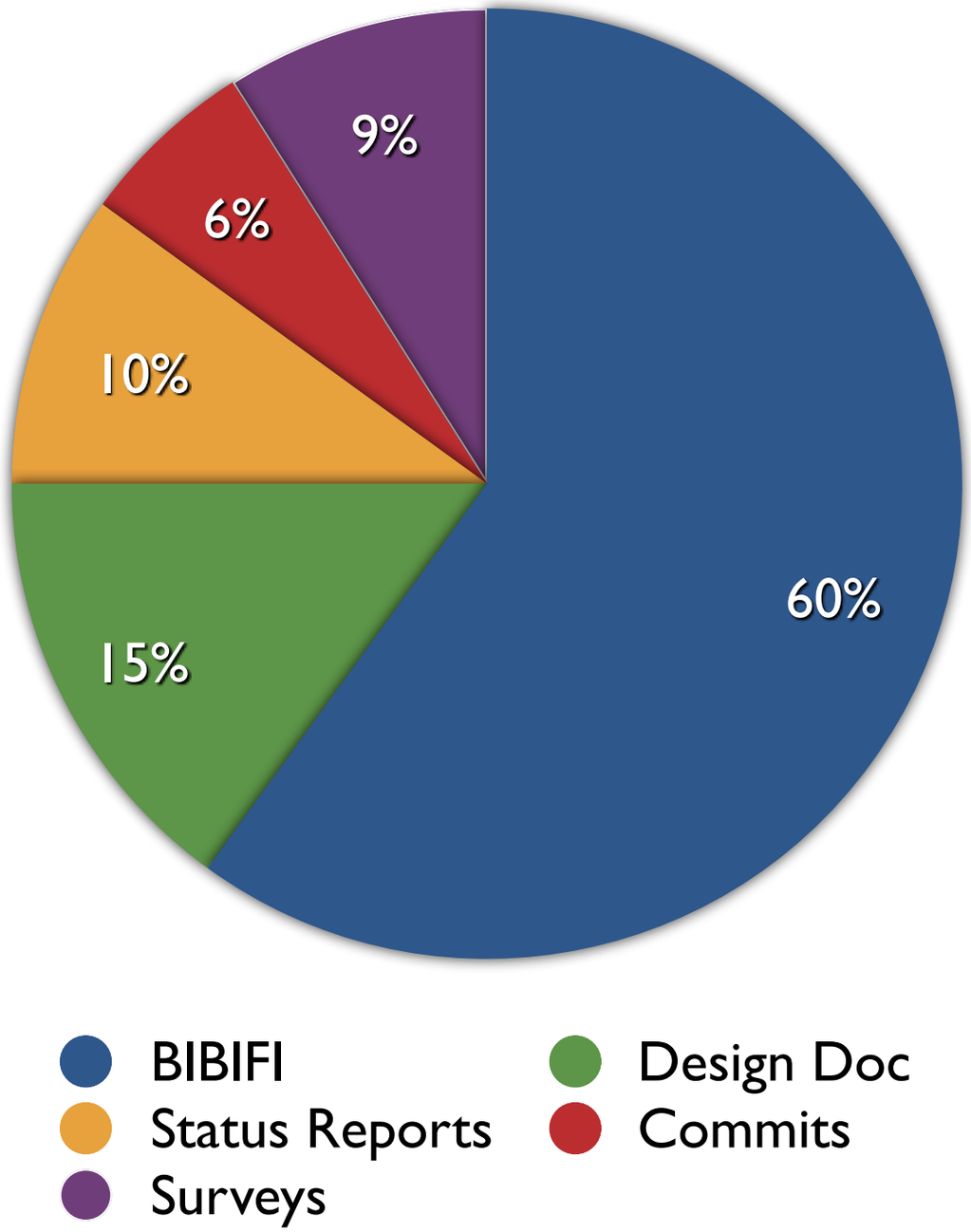
- Please notify us as soon as possible!!
  - For foreseeable events, you need to tell us **Today!**

Lecture notes

- Slides will be released on the course web page after each class.
- I like trees.

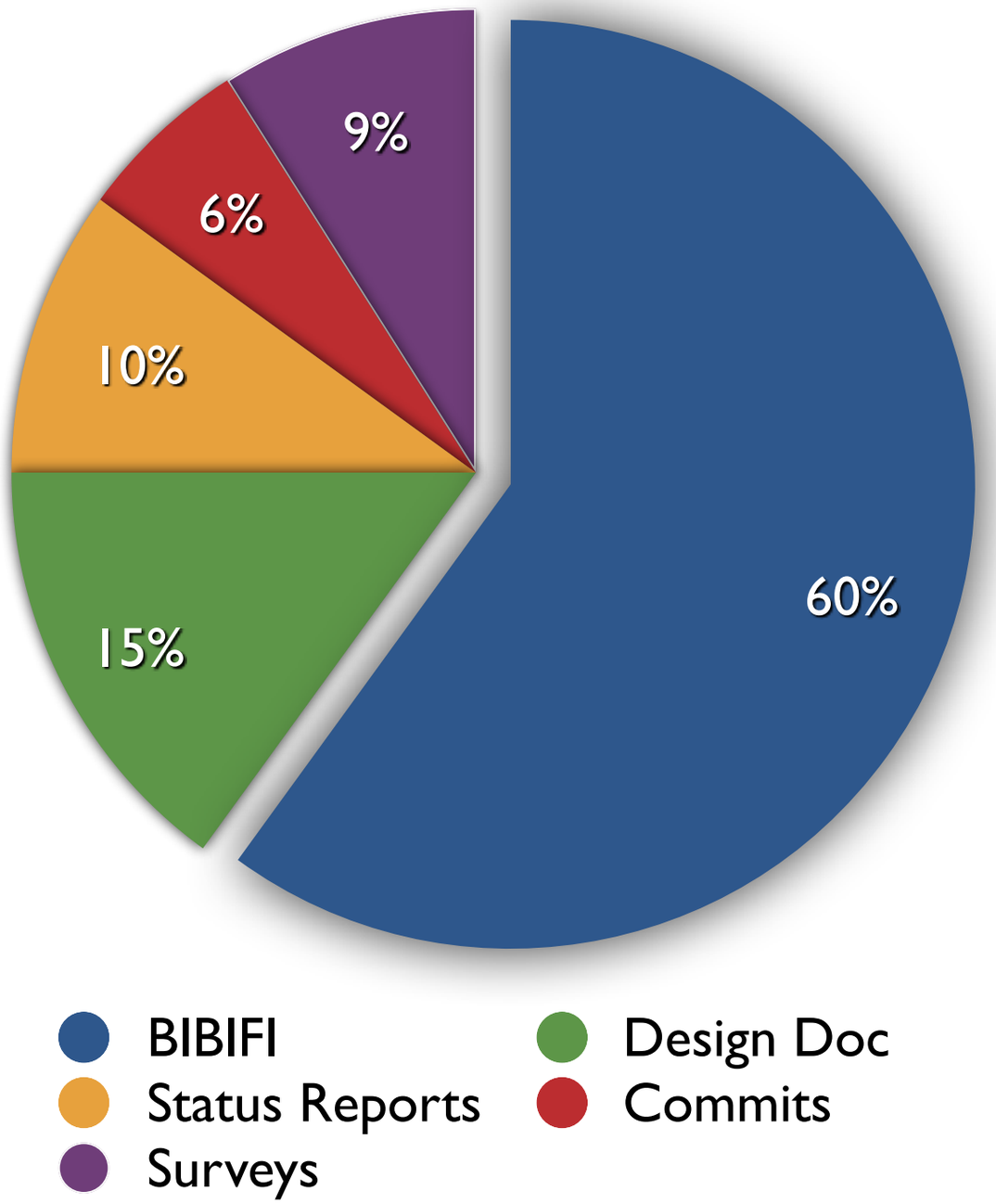


Grading



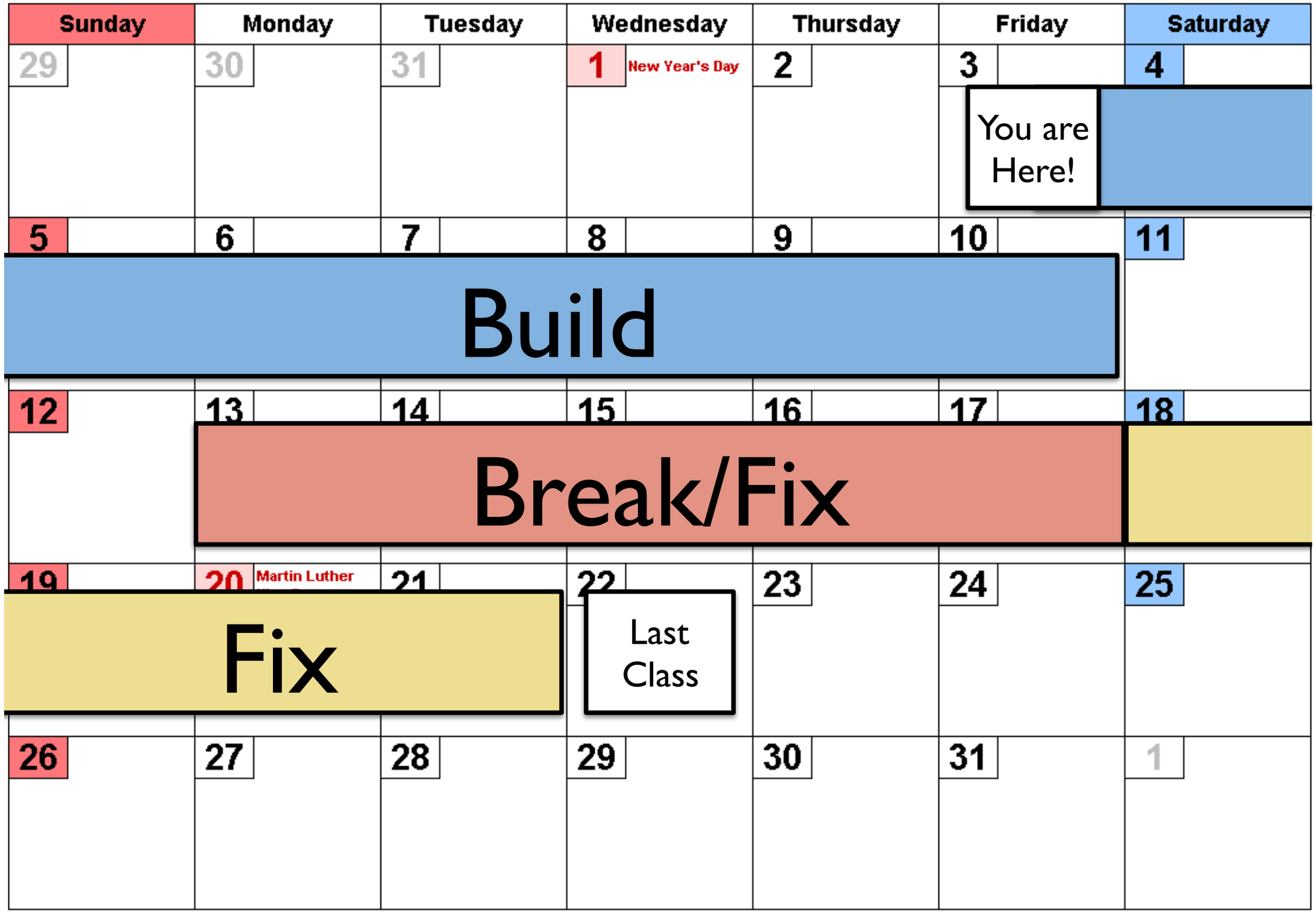
- Course is designed around the BIBIFI project
- No extra credit assignments
- Teams of <u>at most 2</u>

Build It, Break It, Fix It

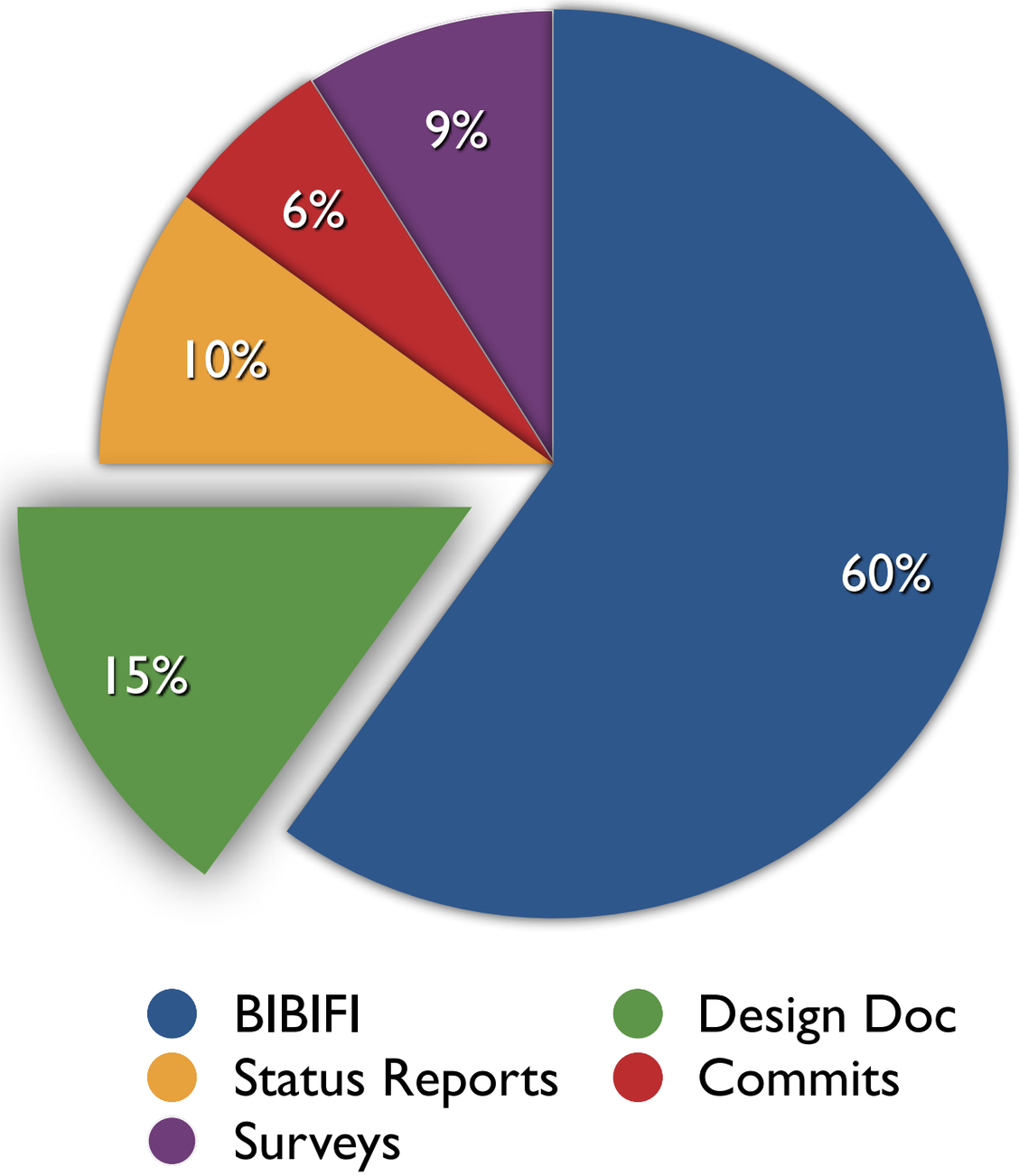


- Majority of the grade is based on the course competition
  - Per-round criterion (40%)
    - Build: Pass all non-optional tests
    - Break: Submit 5 breaks
    - Fix: Fix 50%* of breaks
  - Build and break score ranking (20%)

Build It, Break It, Fix It

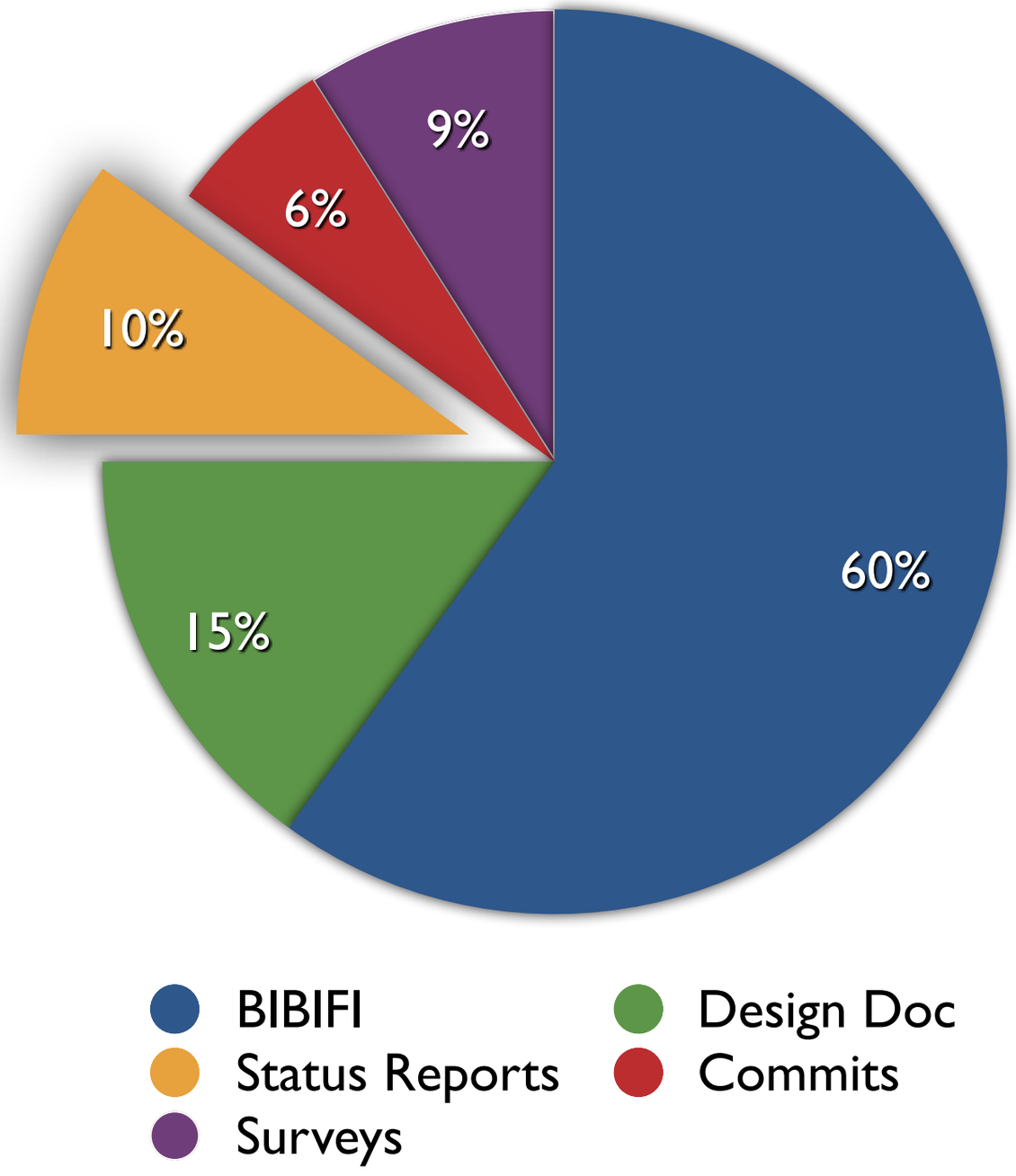


Design Document



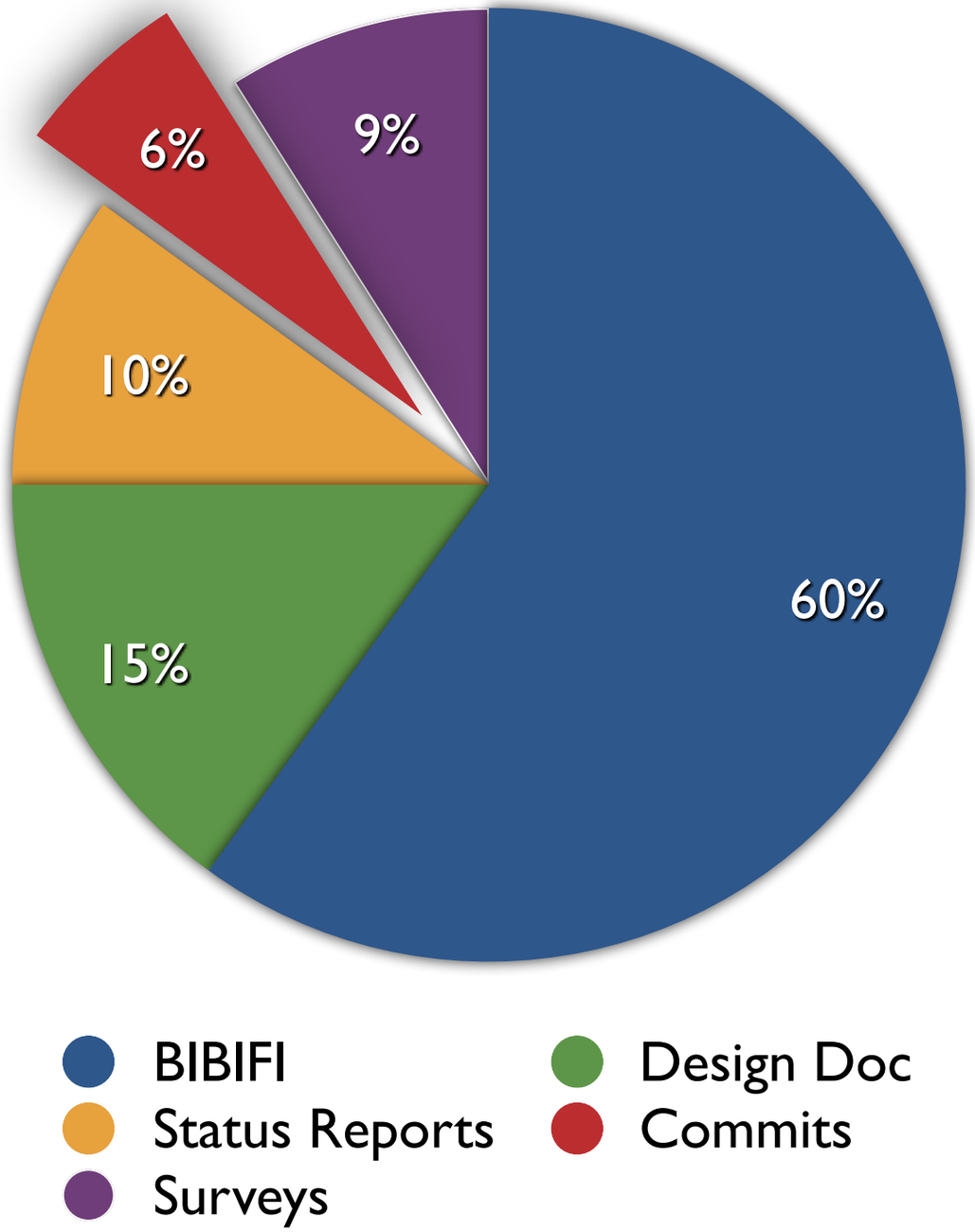
- Answers three questions:
  - How is the program organized?
  - How can an attacker effect the system?
  - How are threats mitigated?
- Three iterations due:
  - Initial design (due **6 Jan**)
  - Build round design (due **13 Jan**)
  - Final design (due **22 Jan**)

Status Reports



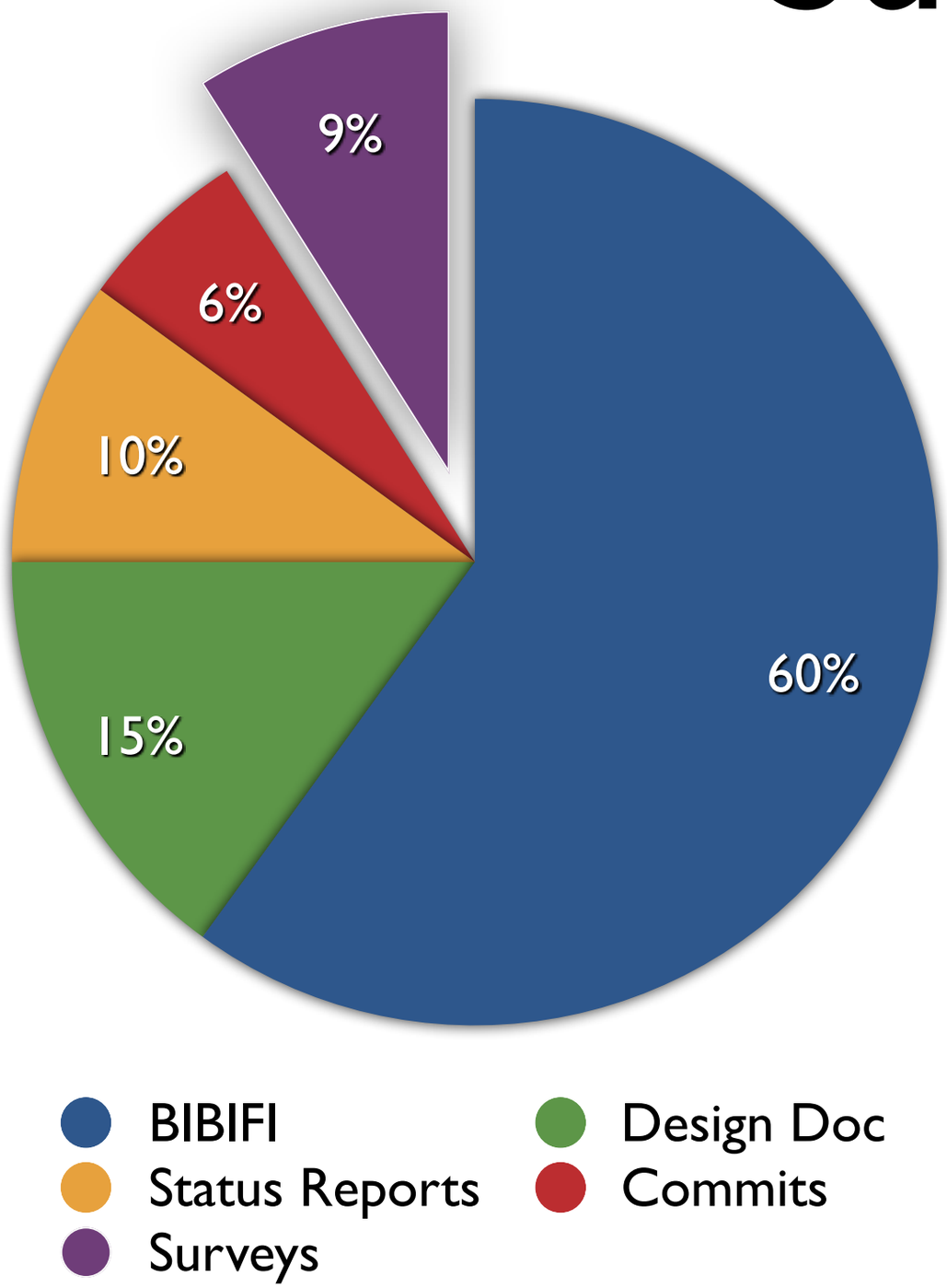
- Submitted every **weekday** individually
- Answer the following questions:
  - What are you currently working on? How do you plan to approach the current problem?
  - Are there any issues you are currently struggling with?
- Submit at ter.ps/388Nreport

Commit Descriptions



- Submitted with every change to the codebase
  - Commits should be made for individual functionality changes
- Build commit requirements:
  - Description of the change
  - Reason for the change
  - Associated requirement
  - How did you come up with the change?
- To provide longer commit in git, do not specify the "-m" flag when committing

Surveys

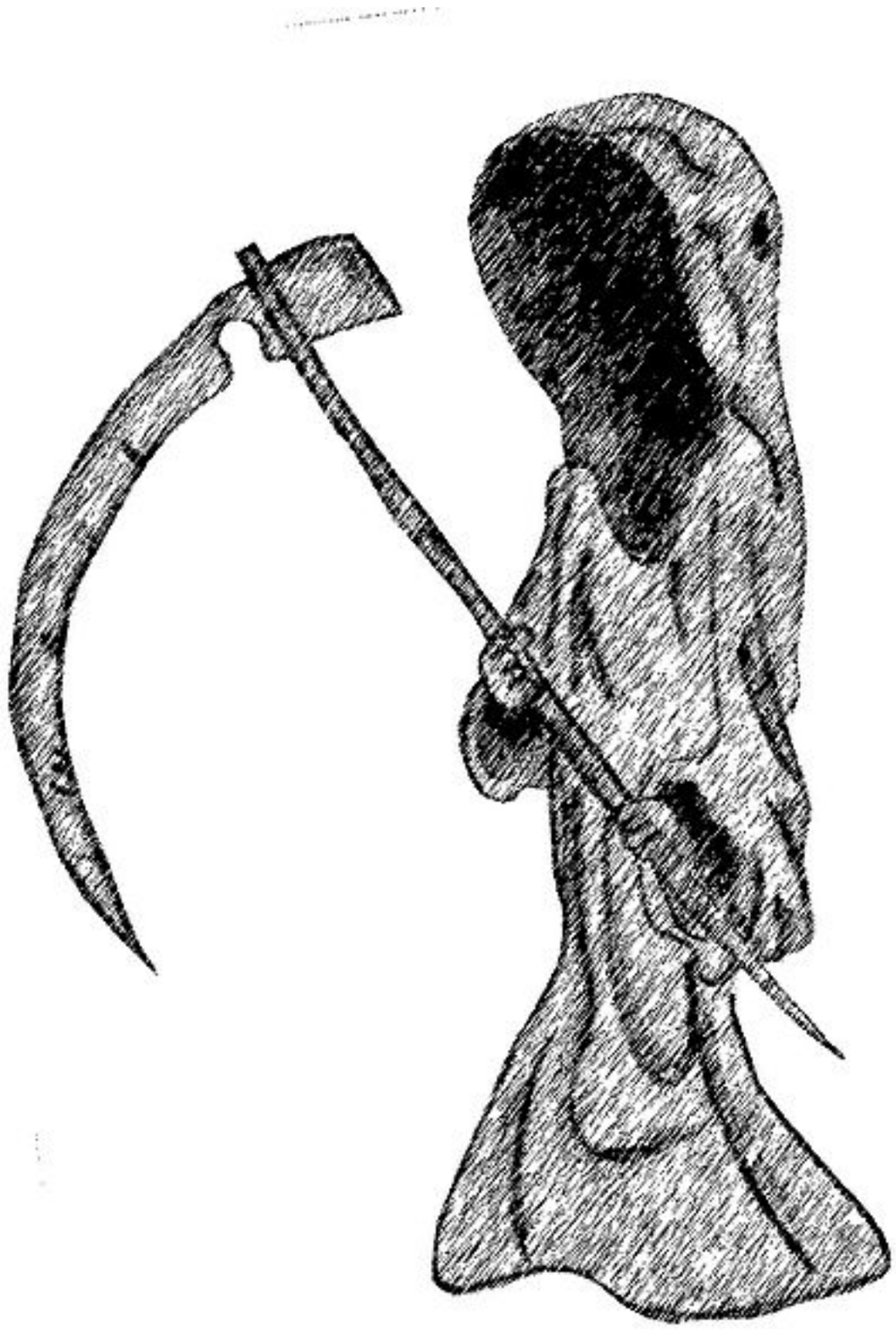


- Three rounds:
  - Pre-course (due by **tonight!**)
  - Mid-course (due **13 Jan**)
  - End-of-course (due **22 Jan**)
- Personalized links for each will be emailed to you

Other Policies

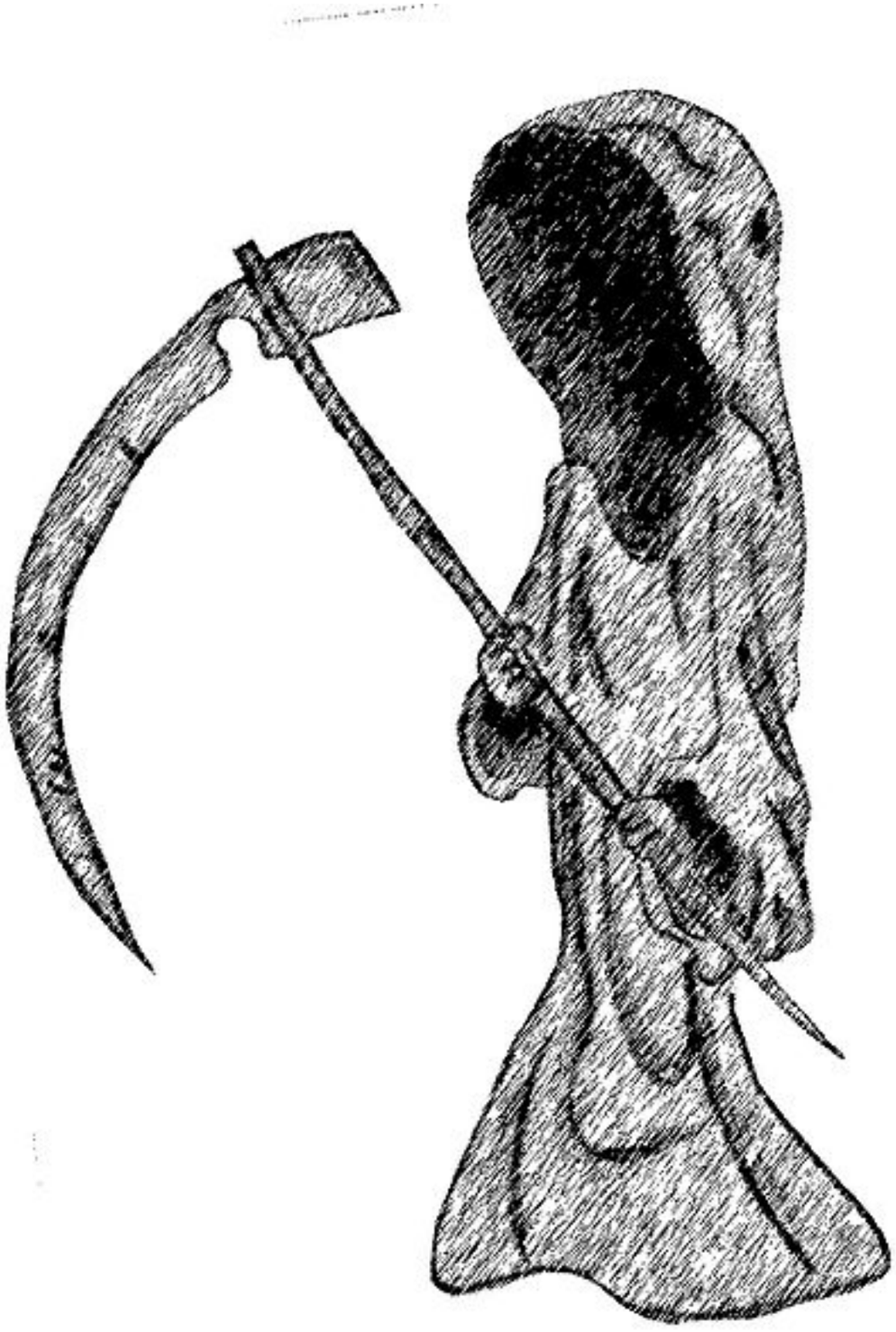
- Please turn off cell phones during class.
- I will do my best to respond to emails and Piazza posts within 24 hours on weekdays (48 hours on weekends).
- Students may appeal to the instructor for reconsideration of a grade, but the appeal must be in writing (i.e., email or private Piazza post), and must be sent within three days (or the close of the semester, whichever is sooner) of receiving the graded assignment.
- Behave civilly: don't be late for class; don't read newspapers/blogs/ etc. during class; don't solve Sudoku puzzles during class; don't struggle with crossword puzzles during class; respect others' opinions.

Cheating policy



- Cheating is not allowed
- We run tools
- If you cheat, you will probably get caught
- **I REFER ALL ACADEMIC DISHONESTY INCIDENTS TO THE HONOR COUNCIL WITHOUT EXCEPTION**
- If you are found to be in violation, you will almost certainly get an F on the course (not just for the parts you were caught cheating)
- If you don't cheat and **work hard**, you will always do better than if you cheated

Cheating policy



- Cheating is (but is not limited to):
  - Working together to solve assignment problems (except for group-based assignments)
- Taking credit for something that you did not create
  - It's ok to copy/paste code you found online, but cite it in your comments.

Ethics and Legality

You will learn about, implement attacks:

- Do not use them without explicit written consent from everyone involved!
  - Make sure you know who is involved
- If you want to try something, tell me and I will try to help set up a test environment
- Don't violate: ethics, UMD policies, state and national laws, good sense

Ethics and Legality

You will learn about, implement attacks:

- Do not use them without explicit written consent from everyone involved!
  - Make sure you know who is involved
- If you want to try something, tell me and I will try to help set up a test environment
- Don't violate: ethics, UMD policies, state and national laws, good sense

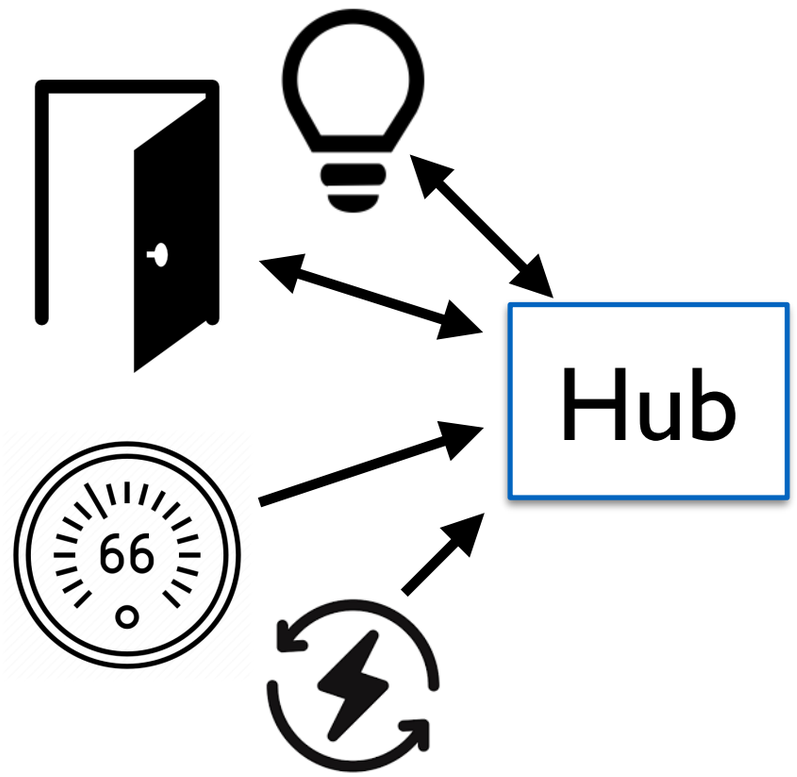
Course credo:

Think like an attacker,

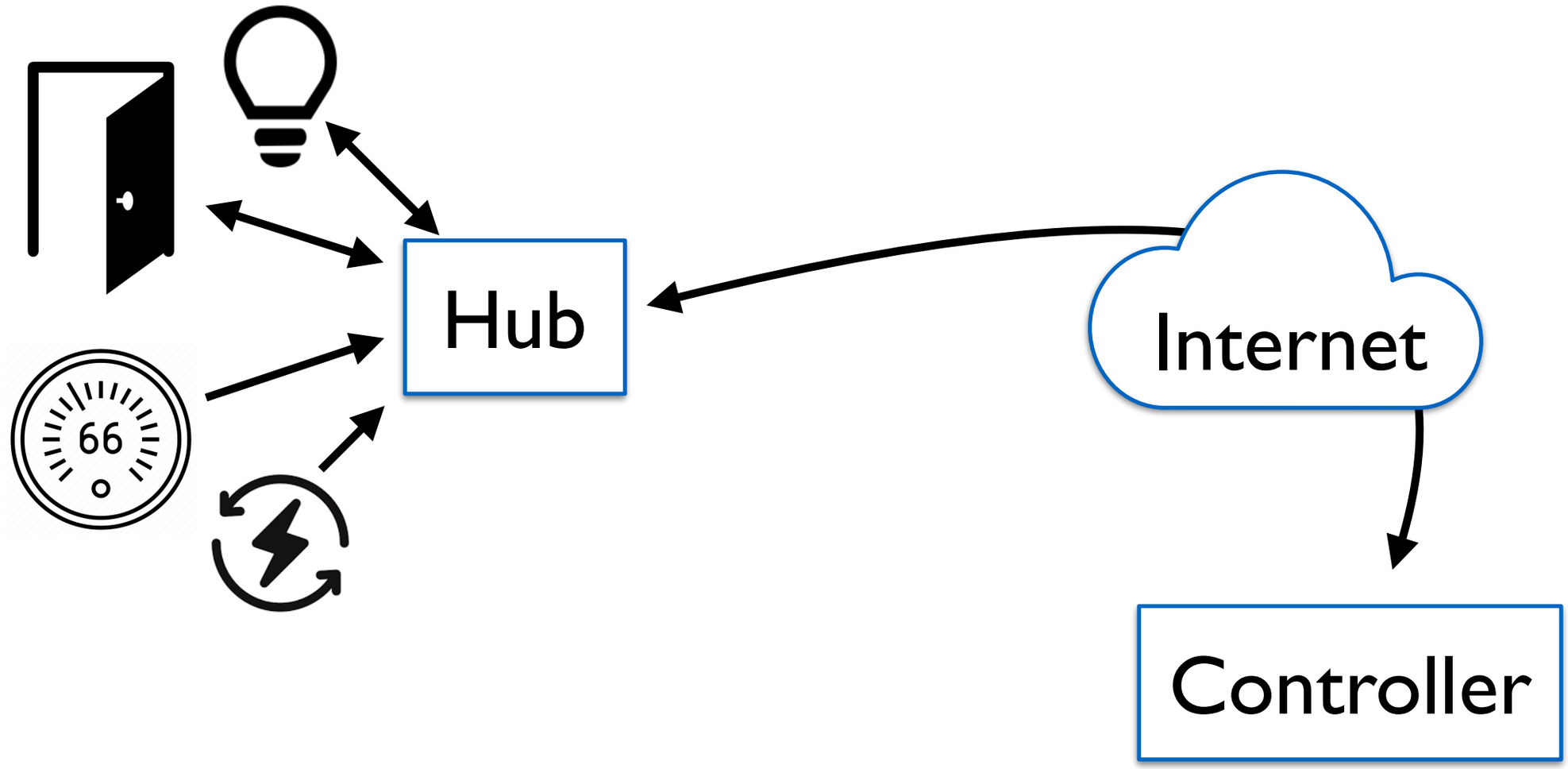
but behave like a responsible adult.

Project Description

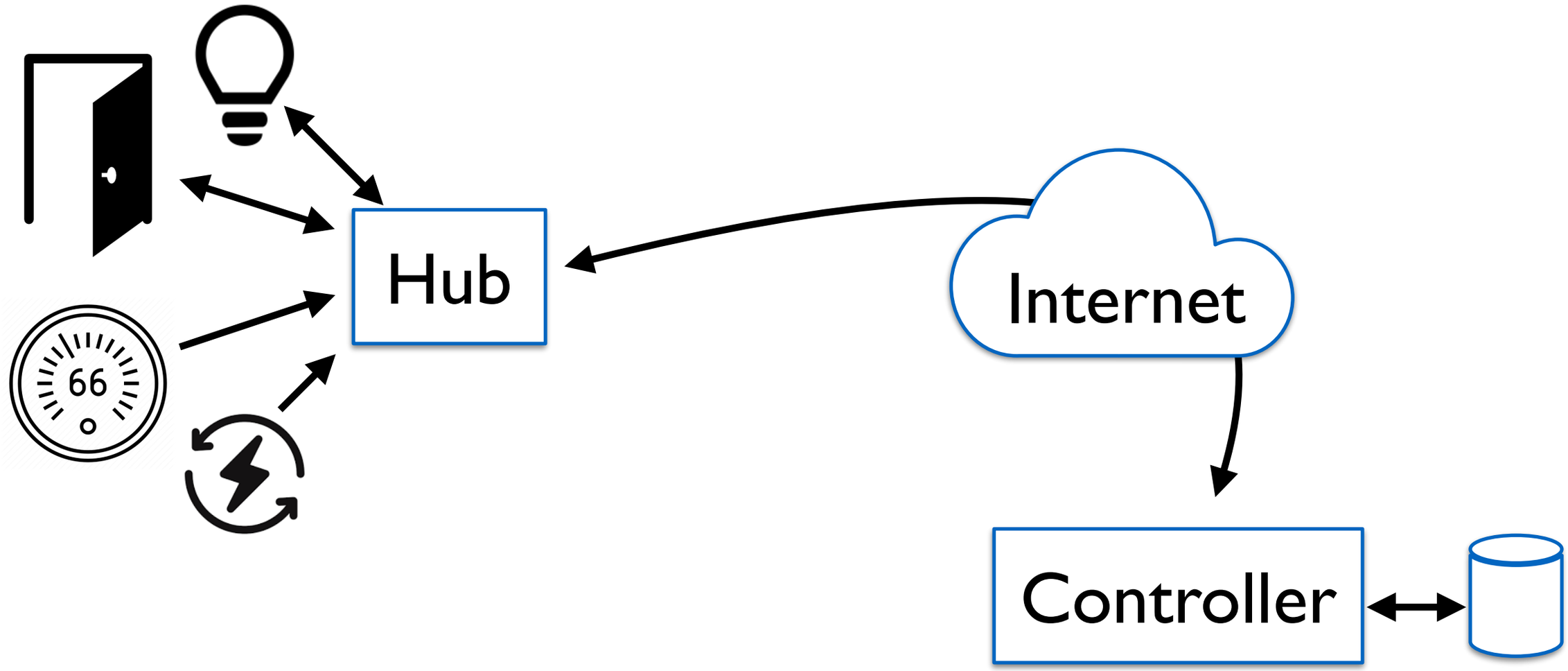
IoT Smart Home



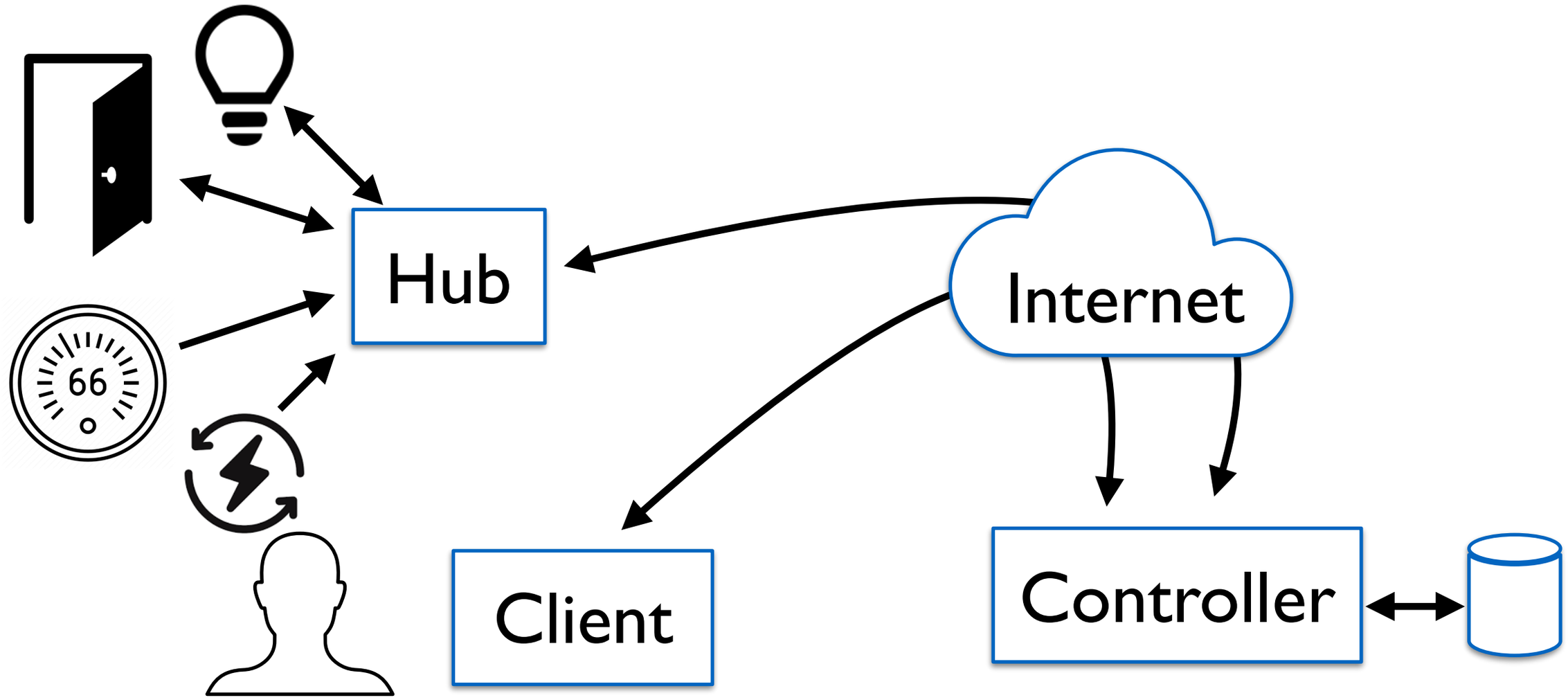
IoT Smart Home



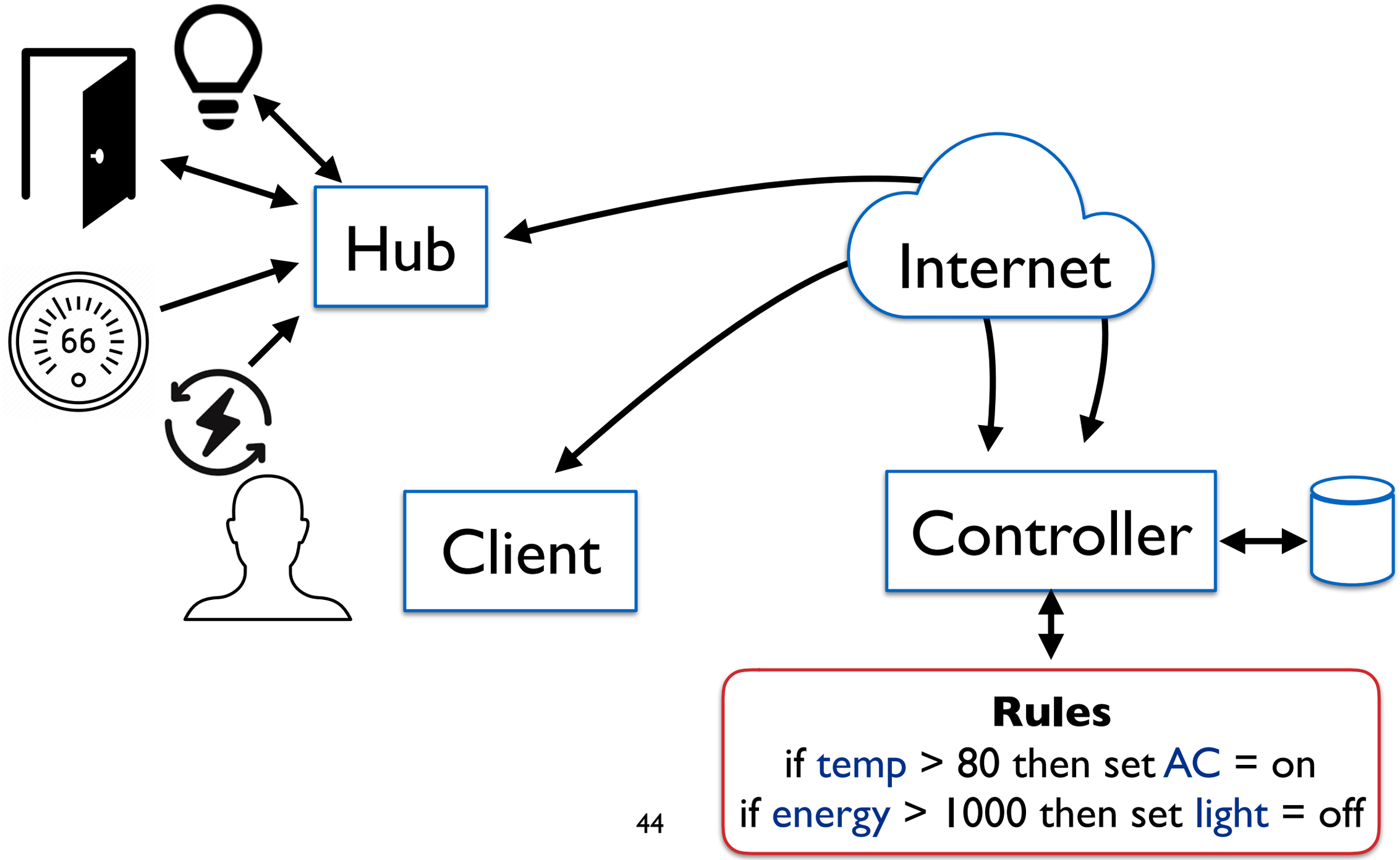
IoT Smart Home



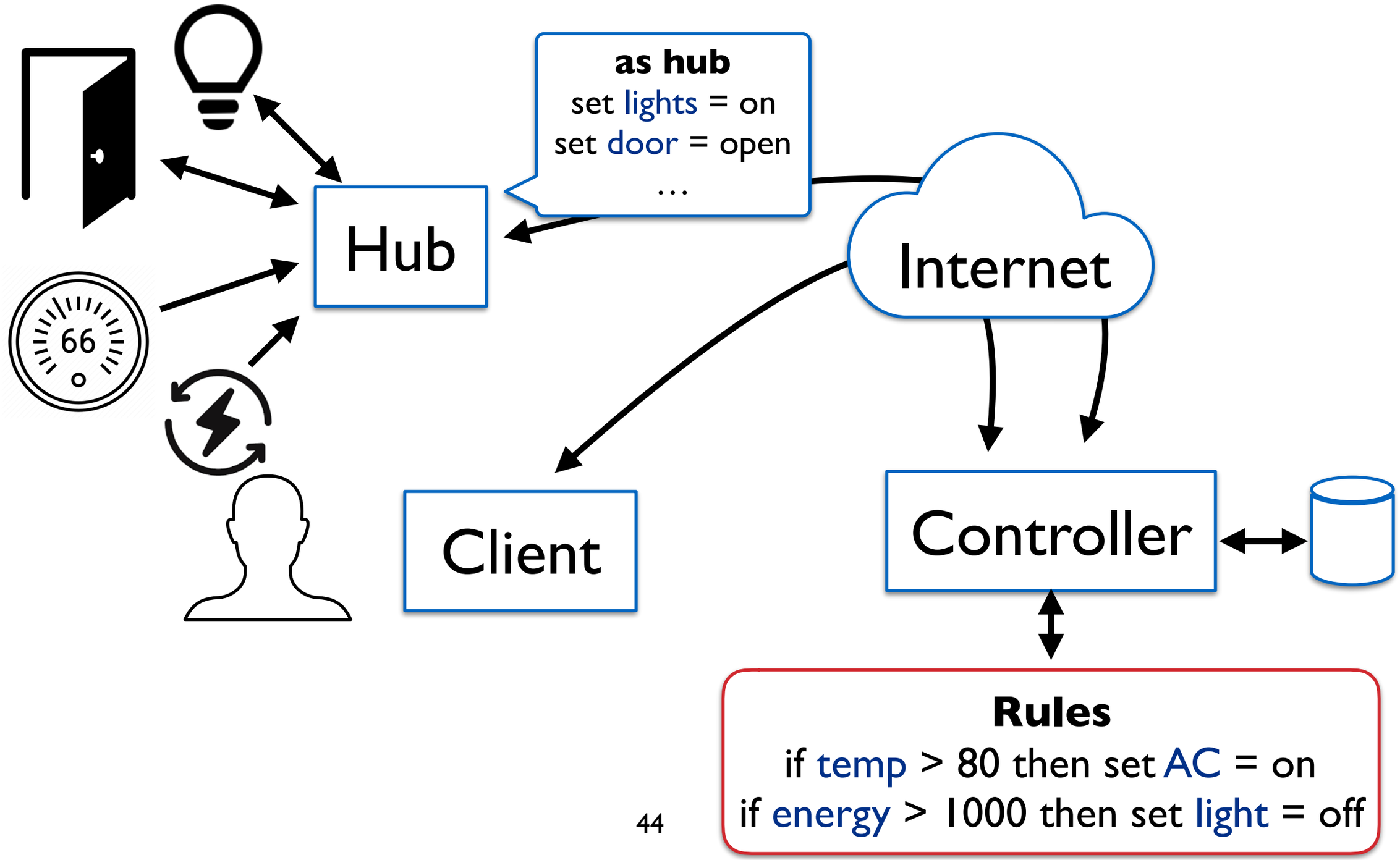
IoT Smart Home



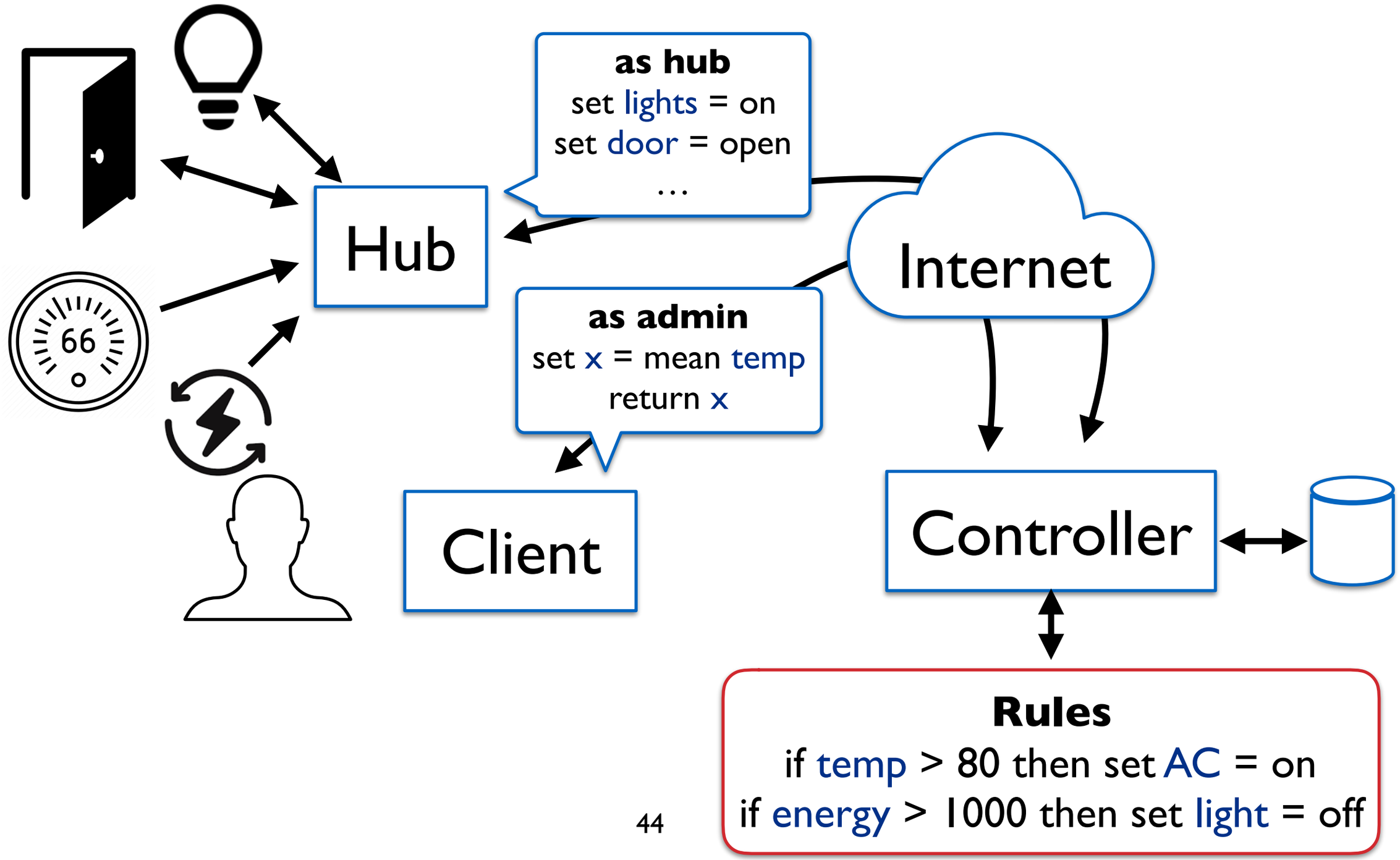
IoT Smart Home



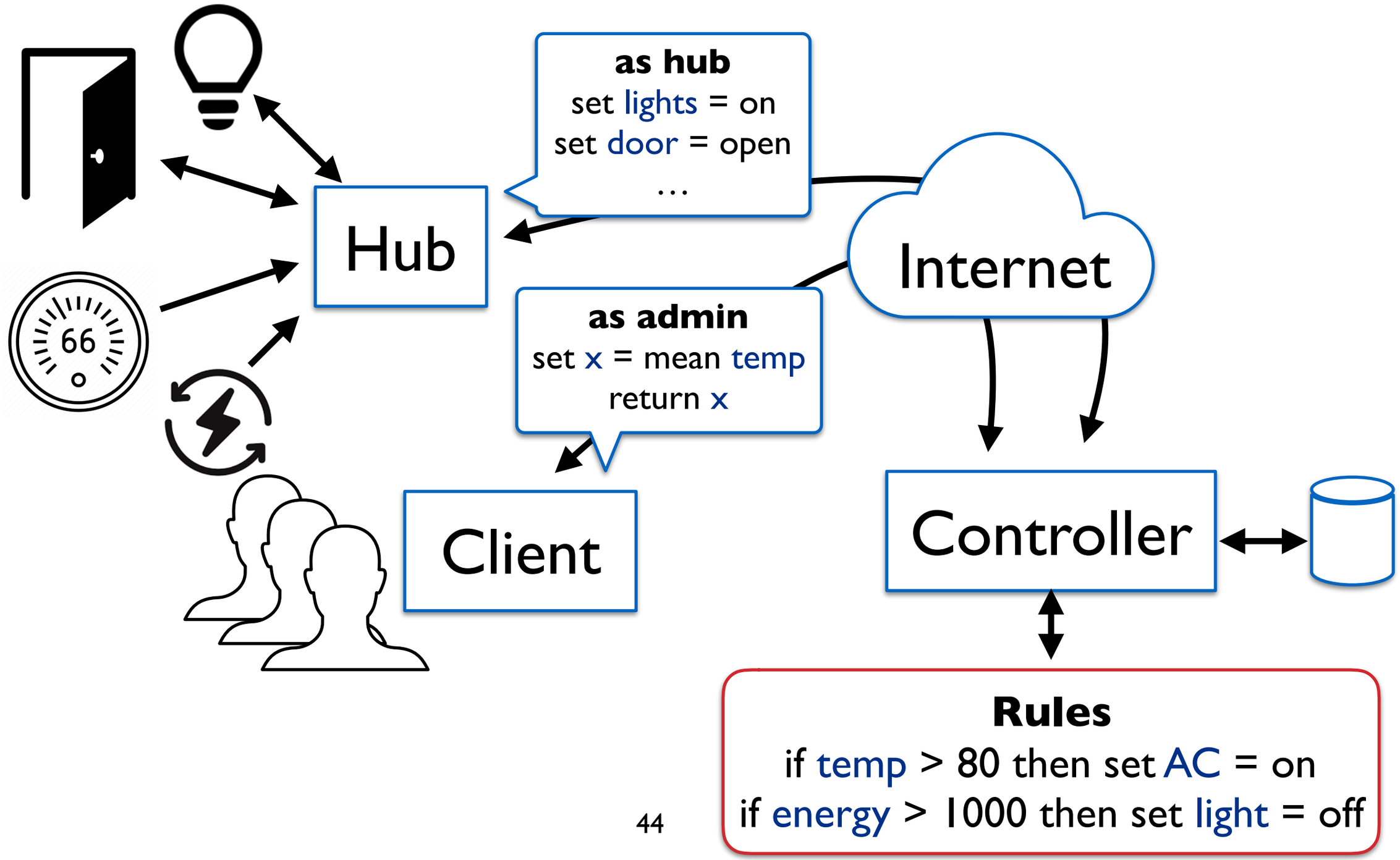
IoT Smart Home



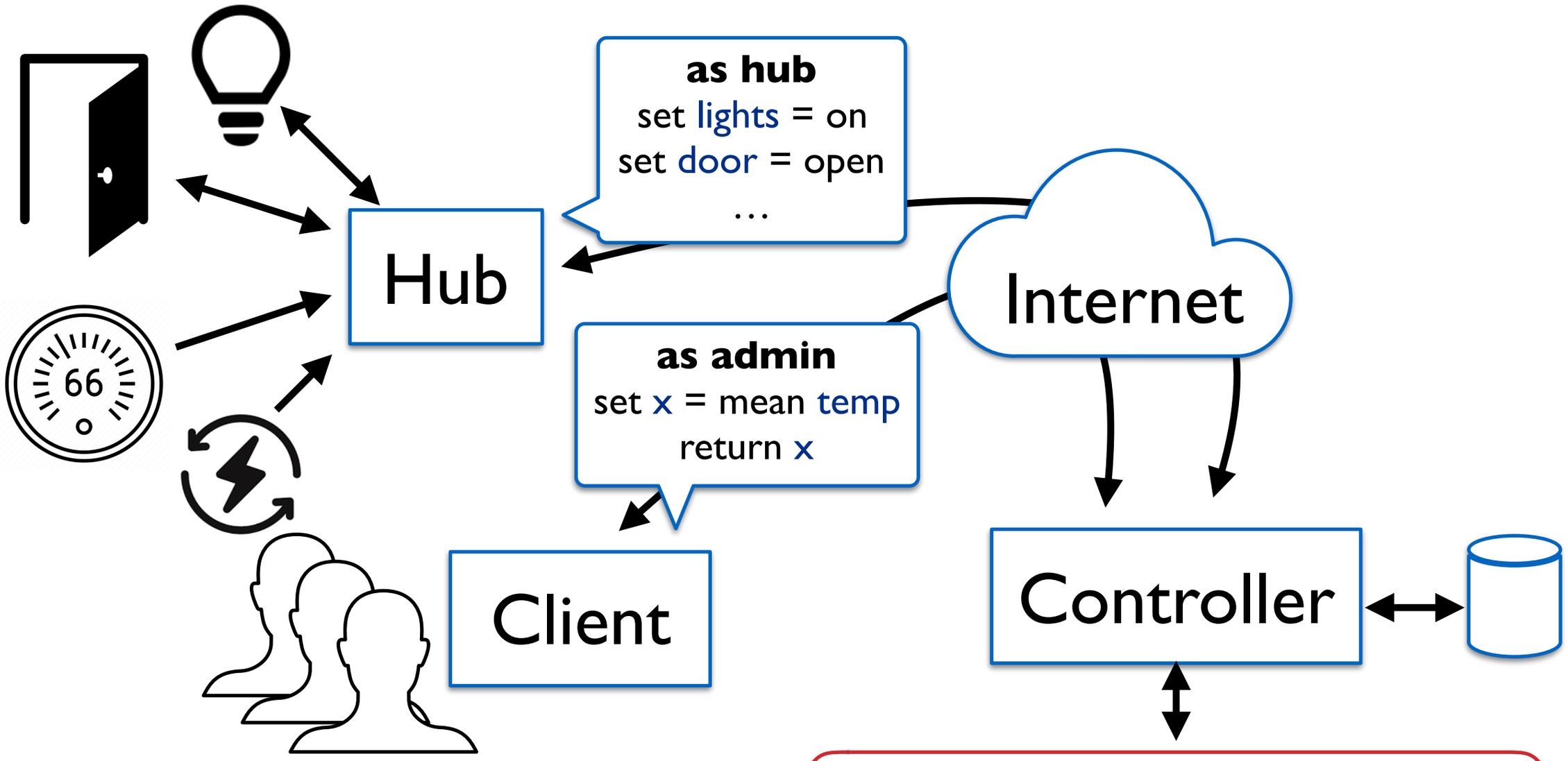
IoT Smart Home



IoT Smart Home

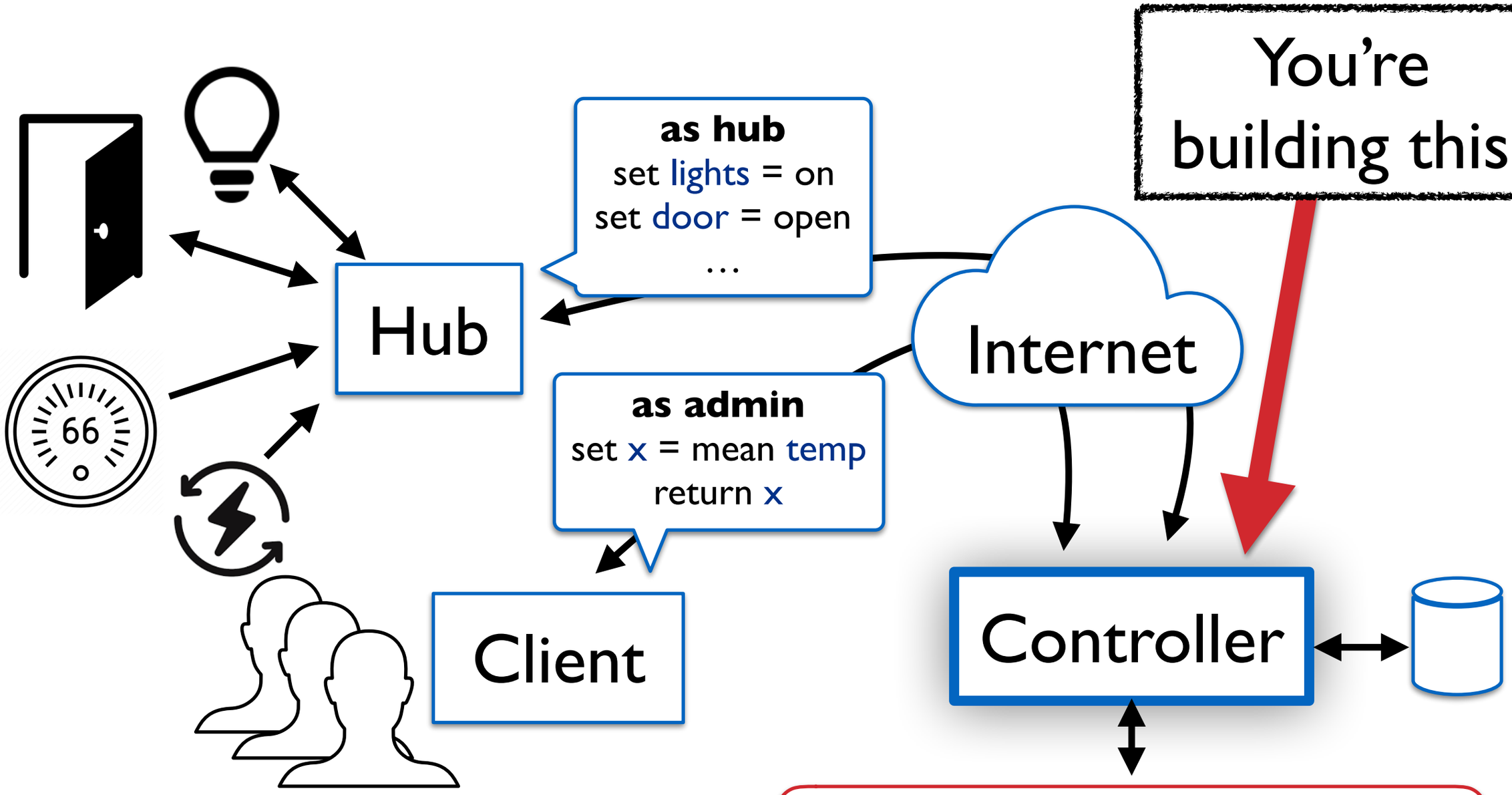


IoT Smart Home



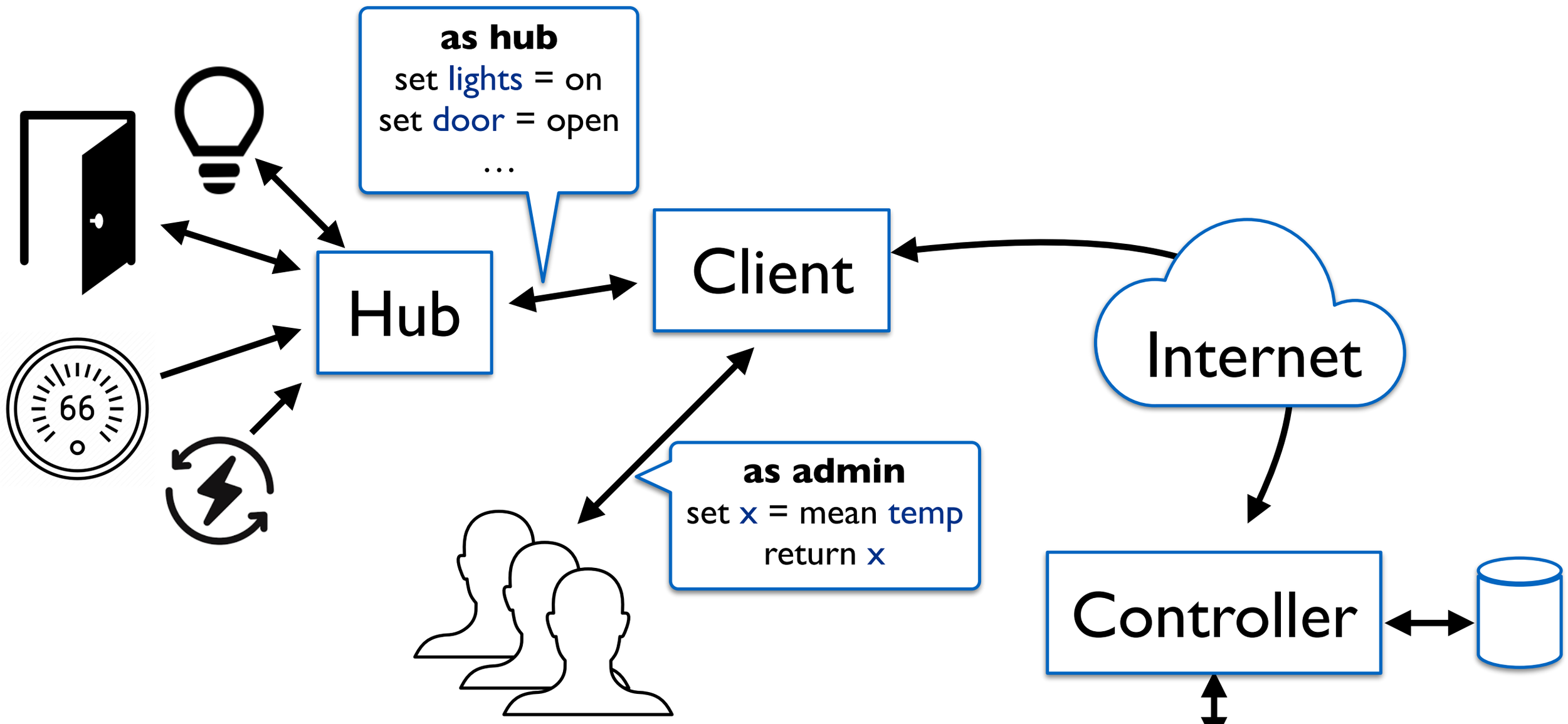
1. Admin can read and write everything*
2. Admin can create new users with limited access
3. Users can delegate their privileges to other users

IoT Smart Home



1. Admin can read and write everything*
2. Admin can create new users with limited access
3. Users can delegate their privileges to other users

IoT Smart Home



1. Admin can read and write everything*
2. Admin can create new users with limited access
3. Users can delegate their privileges to other users

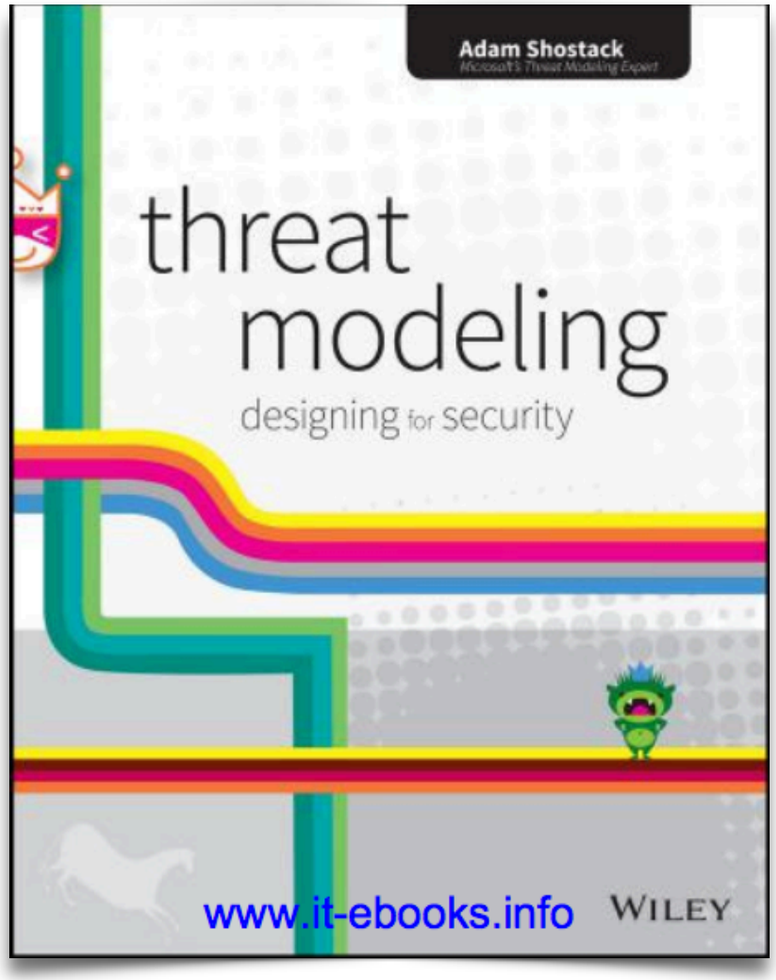
Project Specification

Project Setup

- Register
  - http://128.8.130.12:3000/register
  - Use your directory ID as your username
- Make Teams (due **Tomorrow** by 11:59am)
  - http://128.8.130.12:3000/createteam
  - At most 2 per team
- Setup gitlab
  - gitlab.cs.umd.edu

Threat Modeling

- What are you building?
- What can go wrong?
- What should you do about those things that could go wrong?



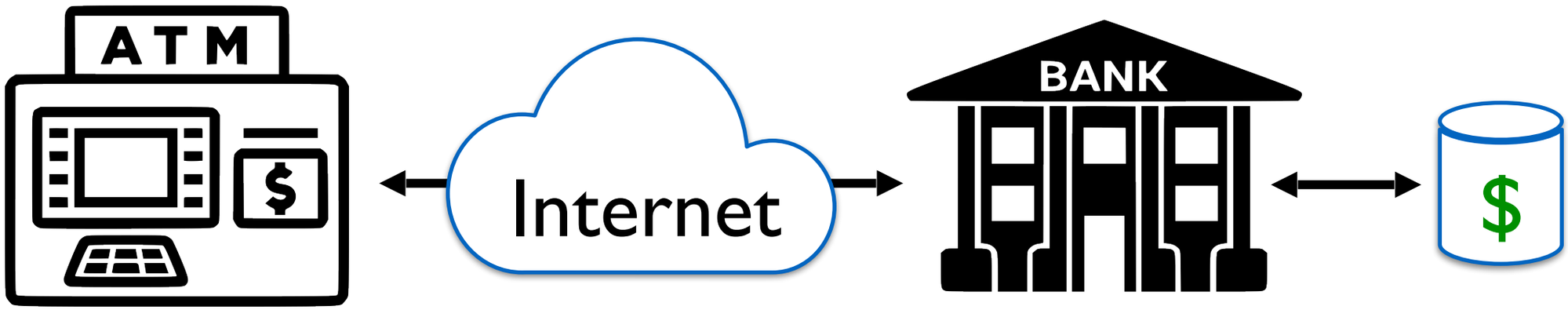
Threat Modeling Homework (Design Doc v1)

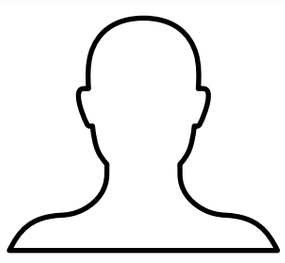
- You'll be identifying possible threats for your IoT system
- Performed individually
- Due January 6th at 11:59am EDT (Monday before class!)
- Detailed instructions on the course website

Threat Modeling Example


CHASE
Choose your bills:
$5 $20 $100
CHASE
CHA

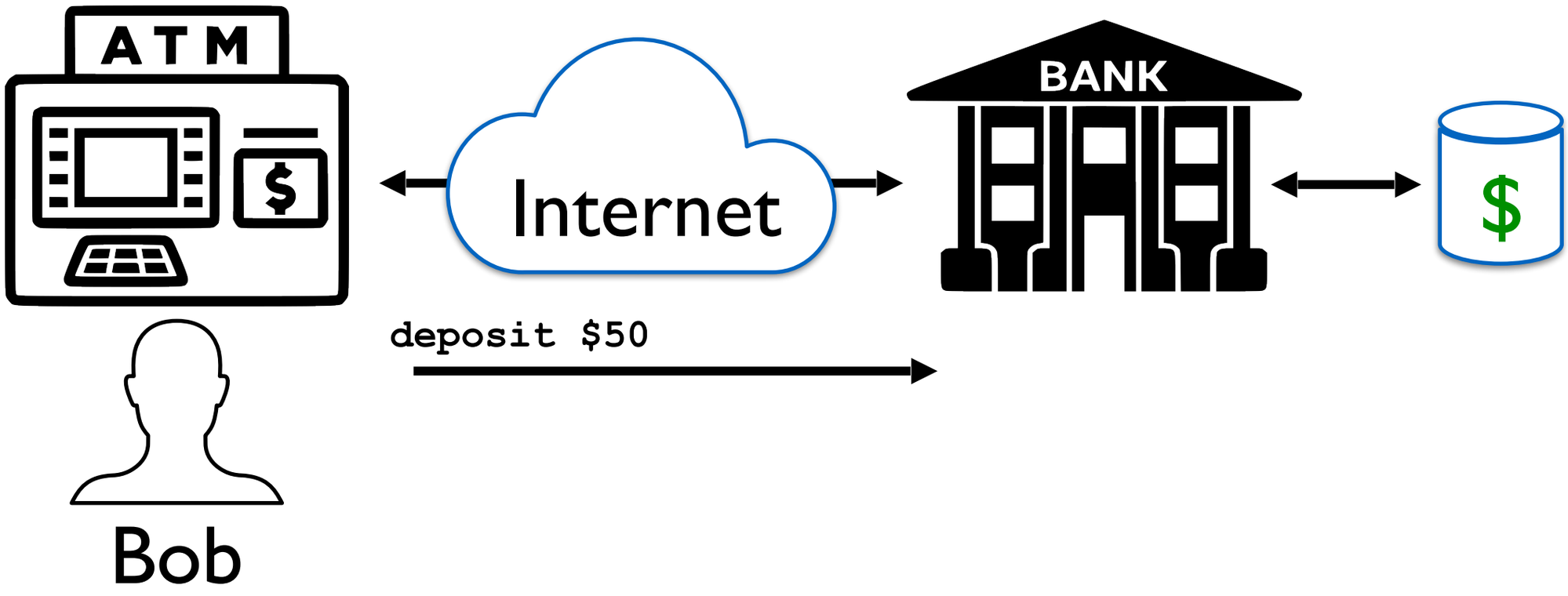
What are you building?



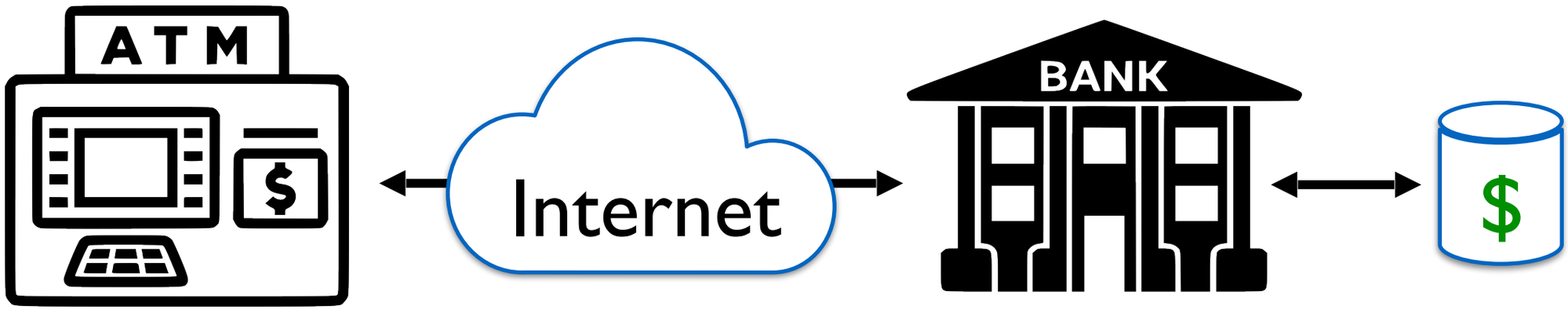


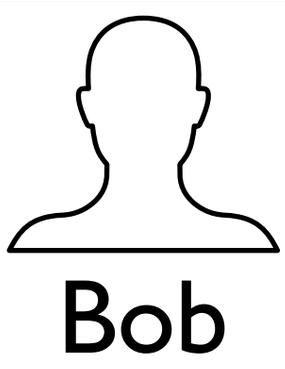
Bob

What are you building?



What are you building?



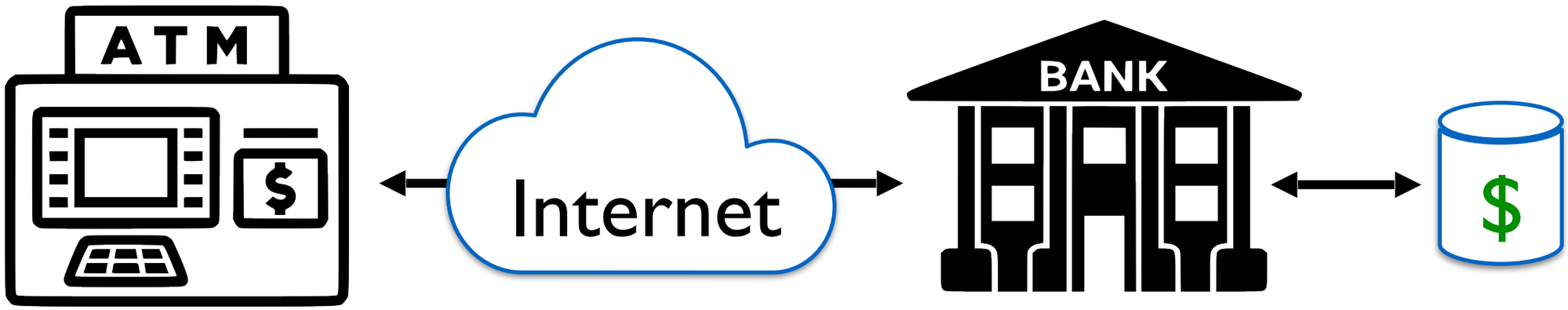


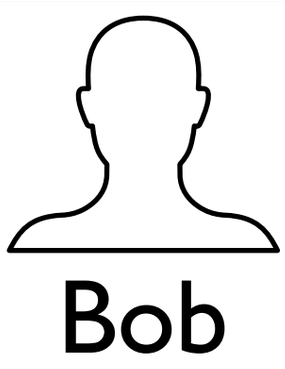
Bob

```
deposit $50
```

bob balance: `1000`

What are you building?



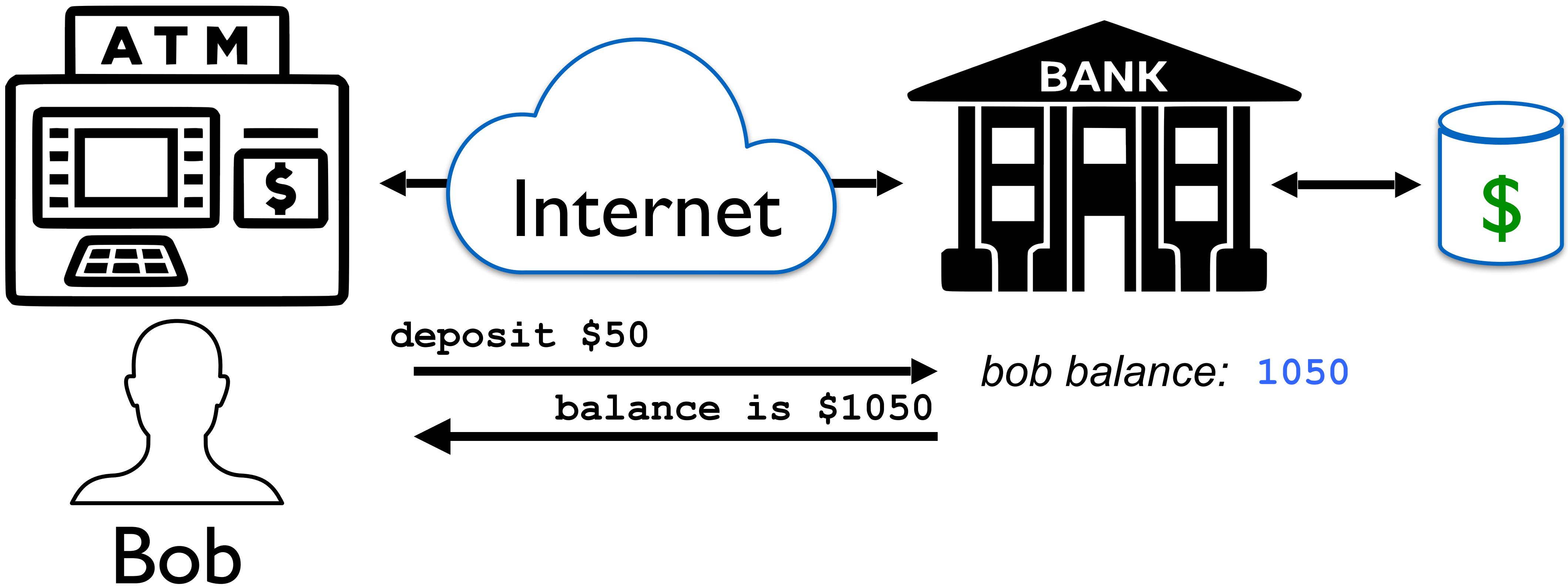


Bob

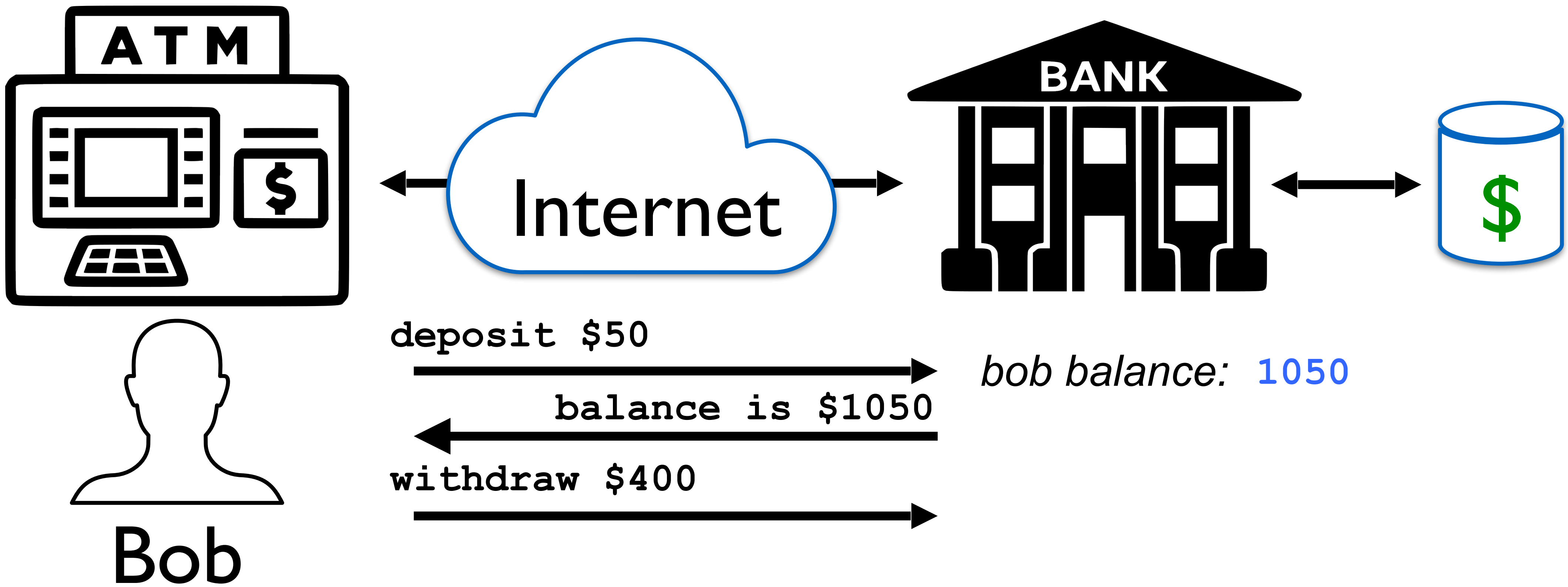
```
deposit $50
```

bob balance: `1050`

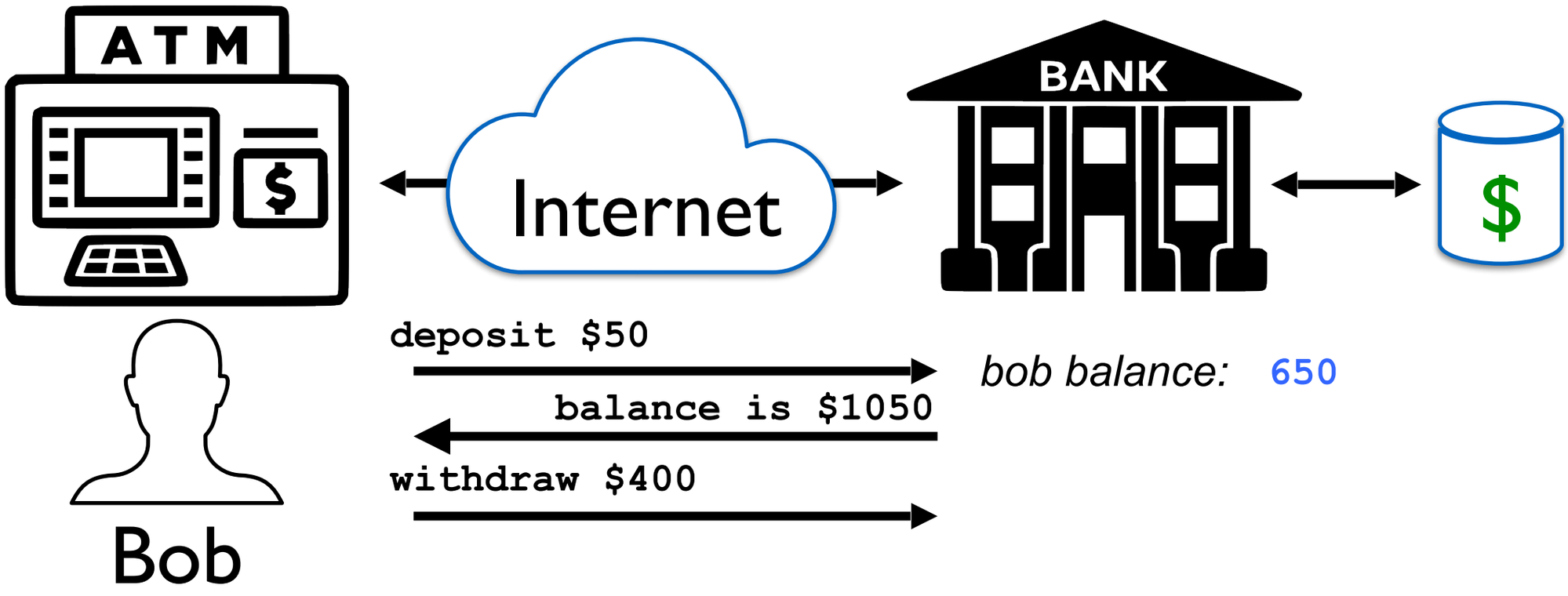
What are you building?



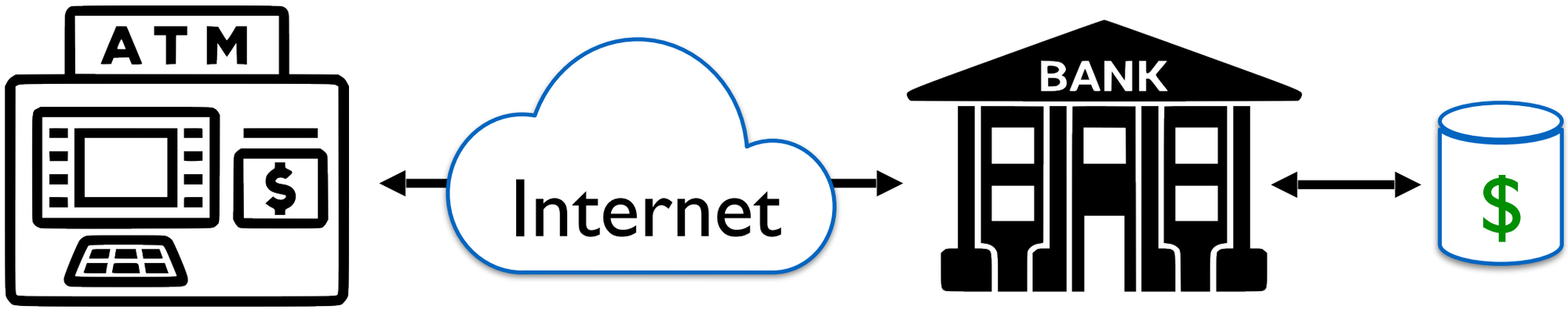
What are you building?

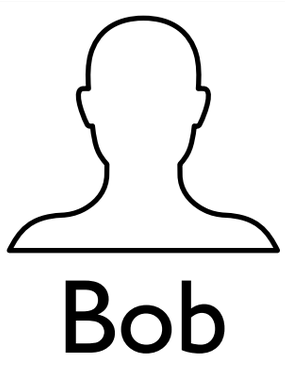


What are you building?



What are you building?

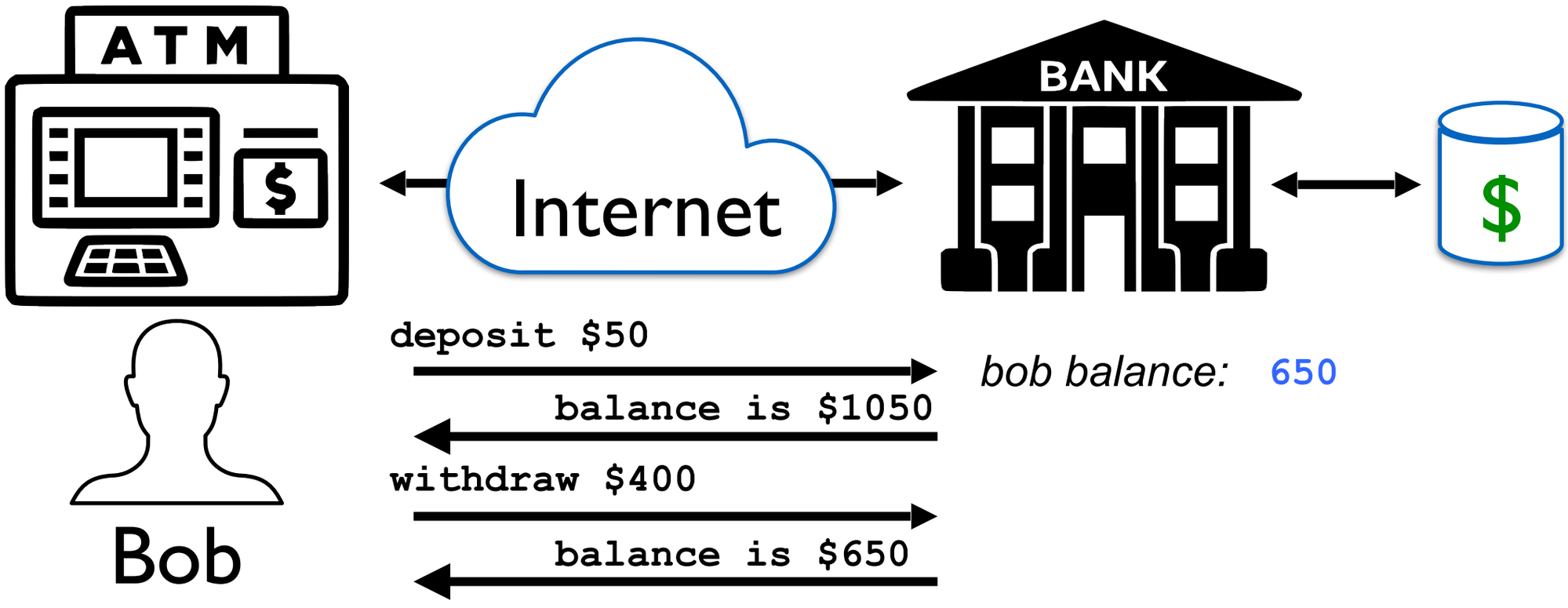




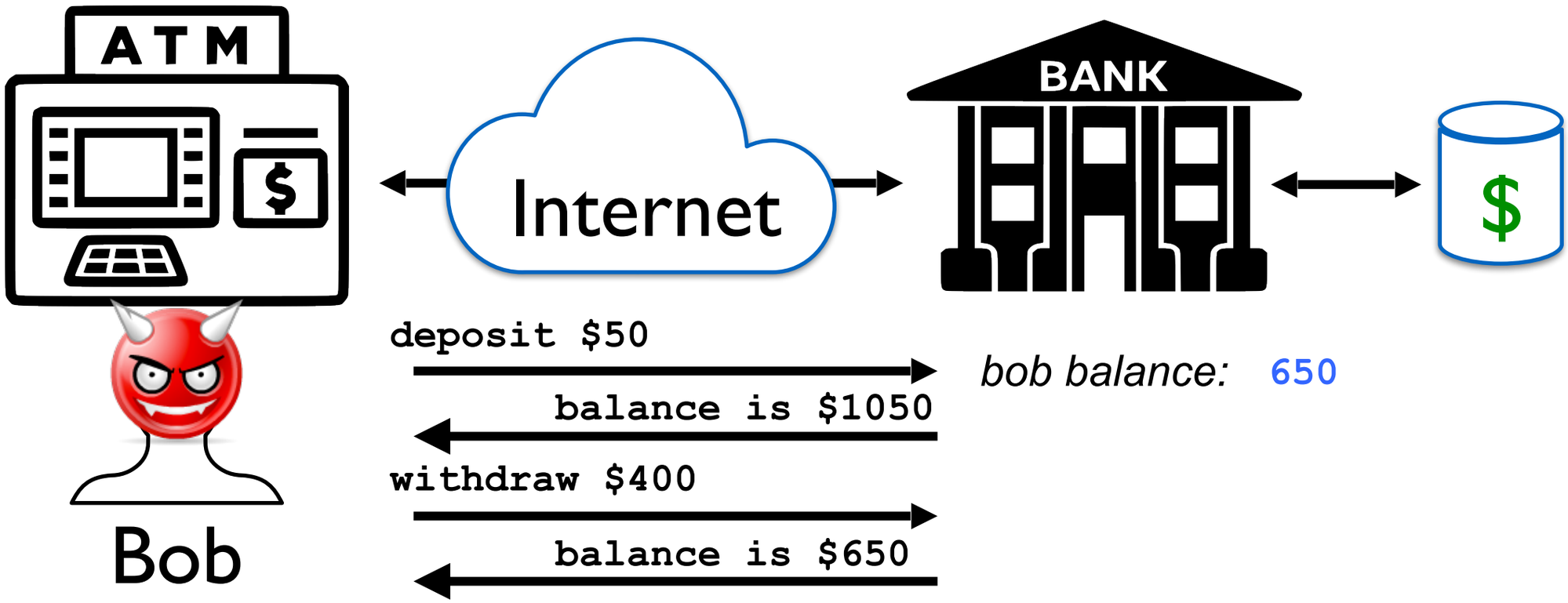
Bob

```
deposit $50
                    bob balance:   650
        balance is $1050
withdraw $400
        balance is $650
```

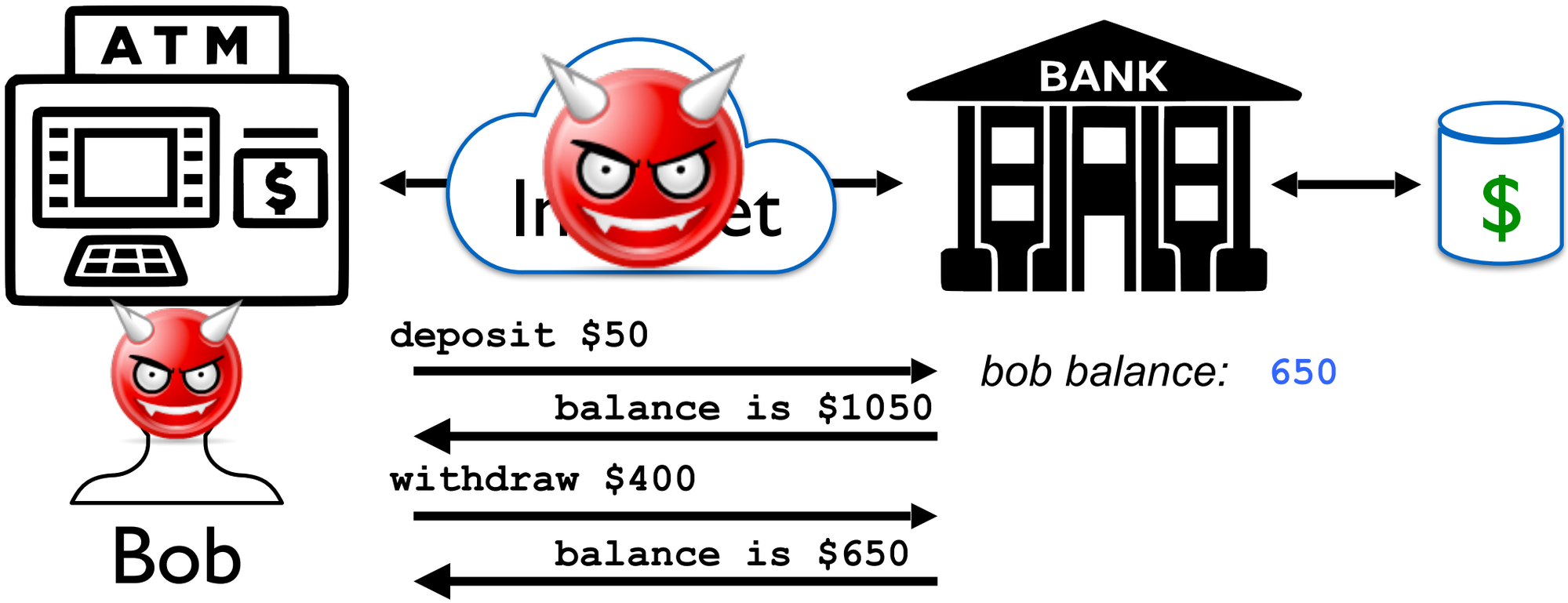
What can go wrong?



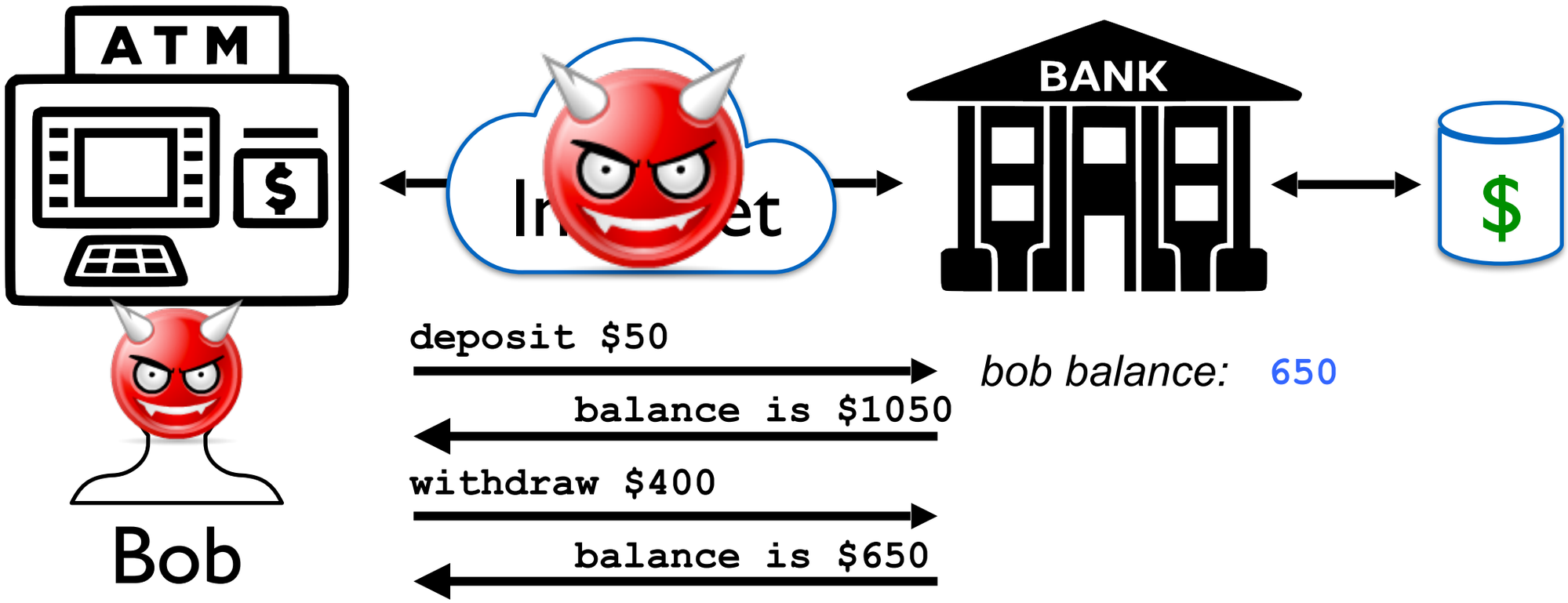
What can go wrong?



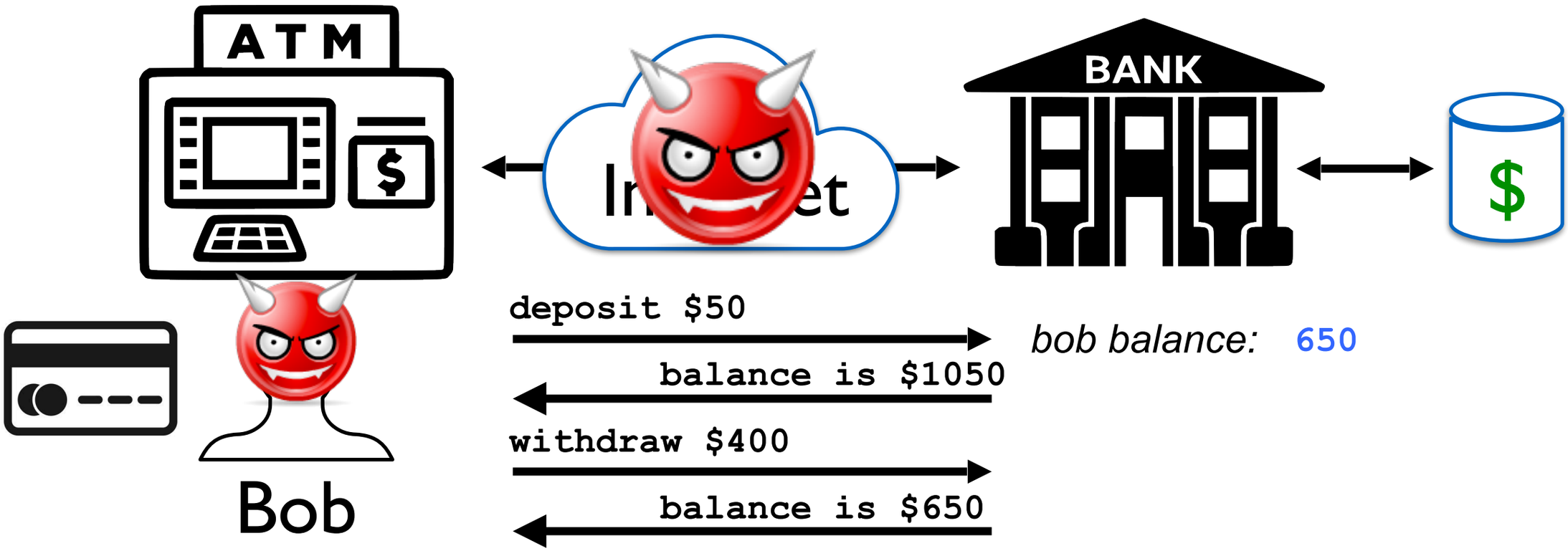
What can go wrong?



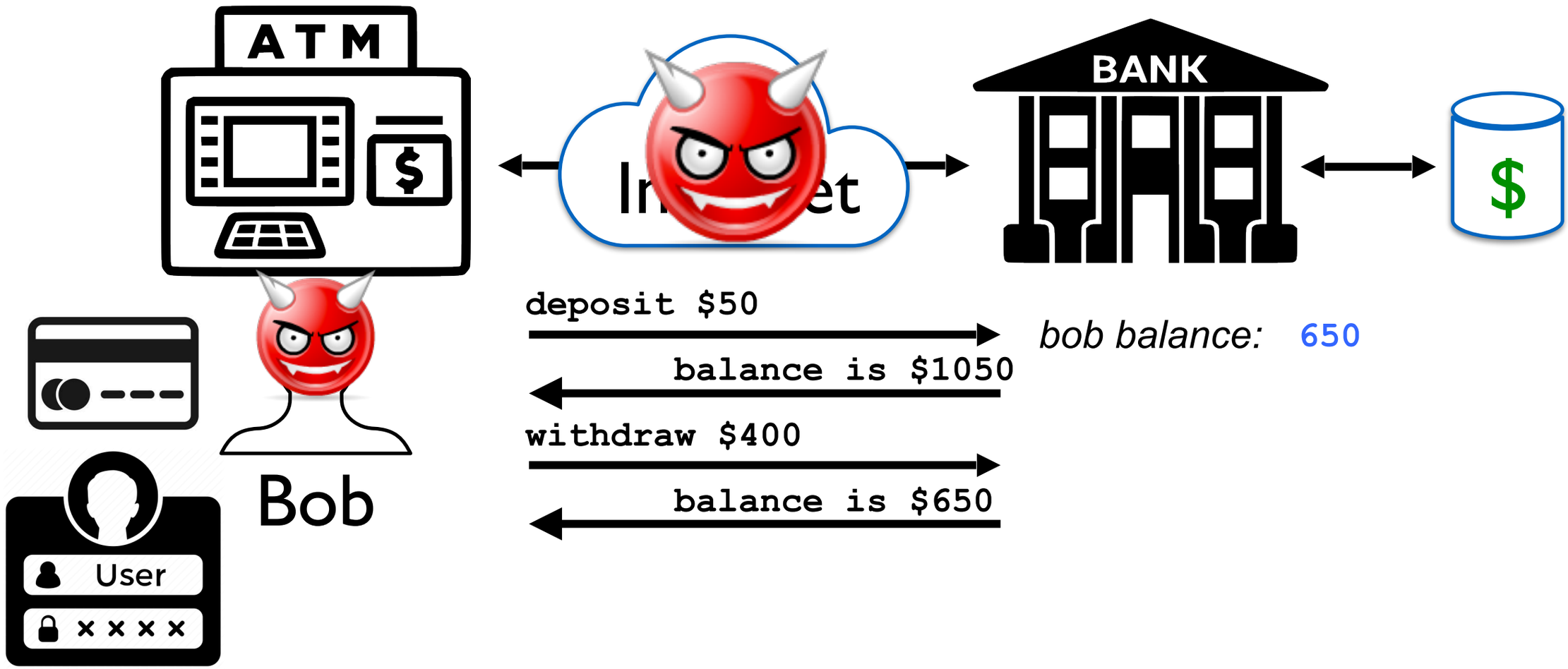
What should we do?



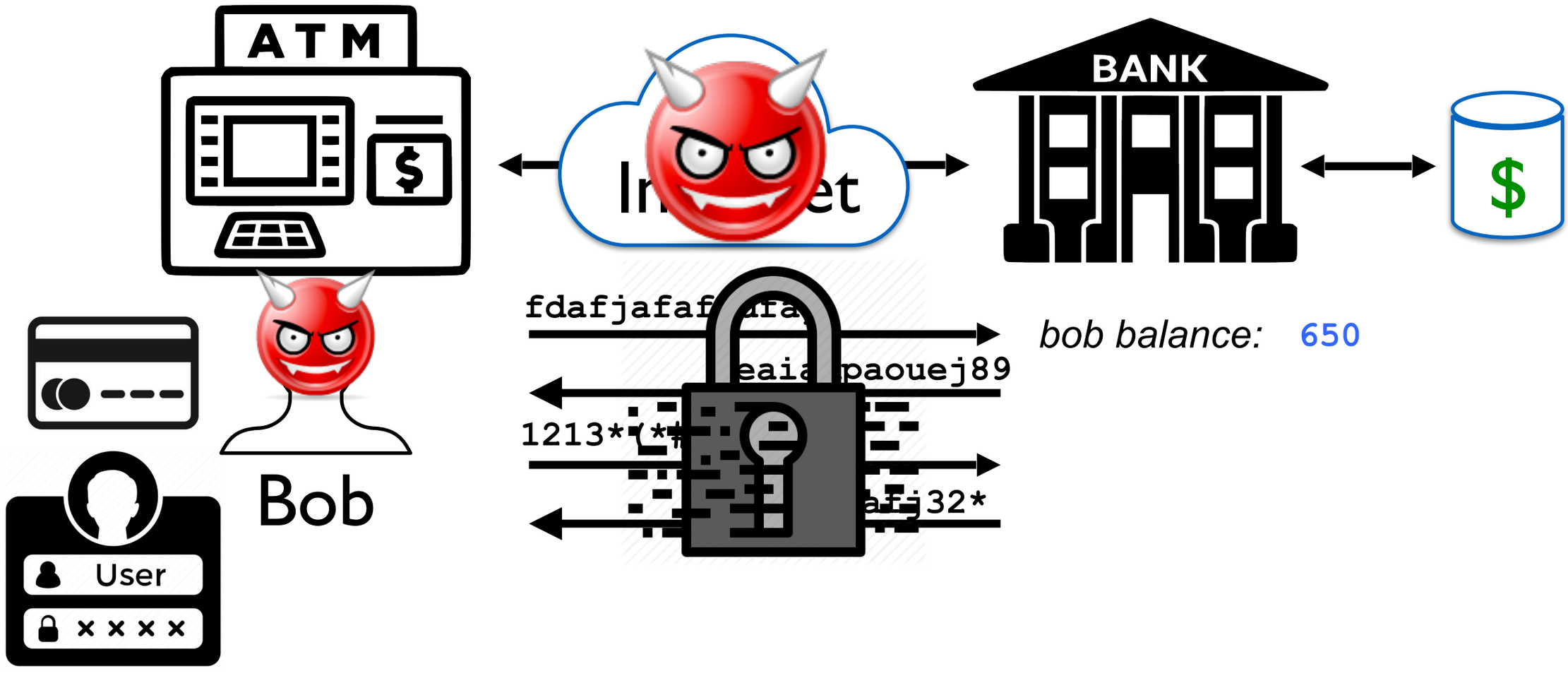
What should we do?



What should we do?



What should we do?



Summary

- Introductions
- Course Overview and Logistics
  - https://www.cs.umd.edu/class/winter2020/cmsc388N/
- IoT Smart Home
- Threat Modeling Example
  - Homework due on **Monday**
- Divide into teams (Due **Saturday** by 11:59am)!
  - http://128.8.130.12:3000/createteam

Summary

- Introductions
- Course Overview and Logistics
  - https://www.cs.umd.edu/class/winter2020/cmsc388N/
- IoT Smart Home
- Threat Modeling Example
  - Homework due on **Monday**
- Divide into teams (Due **Saturday** by 11:59am)!
  - http://128.8.130.12:3000/createteam

Pre-course surveys due **Today**