Deliverables

Blockchain and Distributed Ledger Technologies / # FF

SAPIENZA
UNIVERSITÀ DI ROMA

Claudio Di Ciccio

claudio.diciccio@uniroma1.it
http://www.di.uniroma1.it/~diciccio

Roll call

Computer Science

Cybersecurity
and other programmes excl. CS

Deliverables

- Presentation (optional, to acquire bonus/malus)
  - Showdown on an agreed day at the end of the winter semester
  - Timing: 7’ presentation and live demo + 2’ discussion
  - Slides must be uploaded in the shared folder before the lecture begins
    - Format: Powerpoint or PDF (no time to switch laptops, really)

- Project (mandatory), to be discussed during the oral exam
  - By 16:59 Rome/CET time two days before the exam
    - Right: before tea time
    - No deadline extension
  - Via upload on Google Drive (see the Classroom space)
  - Format: compressed file containing
    - PDF of the report
    - Folder with project files (according to the structure seen in class)
  - Length: Not less than 18 and not more than 30 pages
    - EXCLUDING table of contents and other indices but
    - including references, front page and figures
      - Longer documents are not necessarily better

Presentation

- Students *in the classroom* compose the voting assembly.
- If preferences for a teamwork amount to
  - [85%, 100%] of the assembly → the bonus is +1.5 pts.
  - (66%, 85%) of the assembly → the bonus is +1.25 pts.
  - [33%, 66%] of the assembly → the bonus is +0.75 pts.
  - (15%, 33%) of the assembly → the bonus is +0.25 pt.
  - [0%, 15%] of the assembly → the bonus is 0 pt.
- Not voting is an option (should neither of the projects be deemed worth the bonus).
- The instructor can add or subtract up to 0.5 extra points
  - On top, and independently of, the assembly bonus
- The vote is repeated twice
  1. To assess the technical quality
     - Mastery of the topic, appropriateness of the solution, attention to details
  2. To assess the presentation quality
     - Clarity of exposition, neatness of the message, completeness of the story

Structure of the presentation

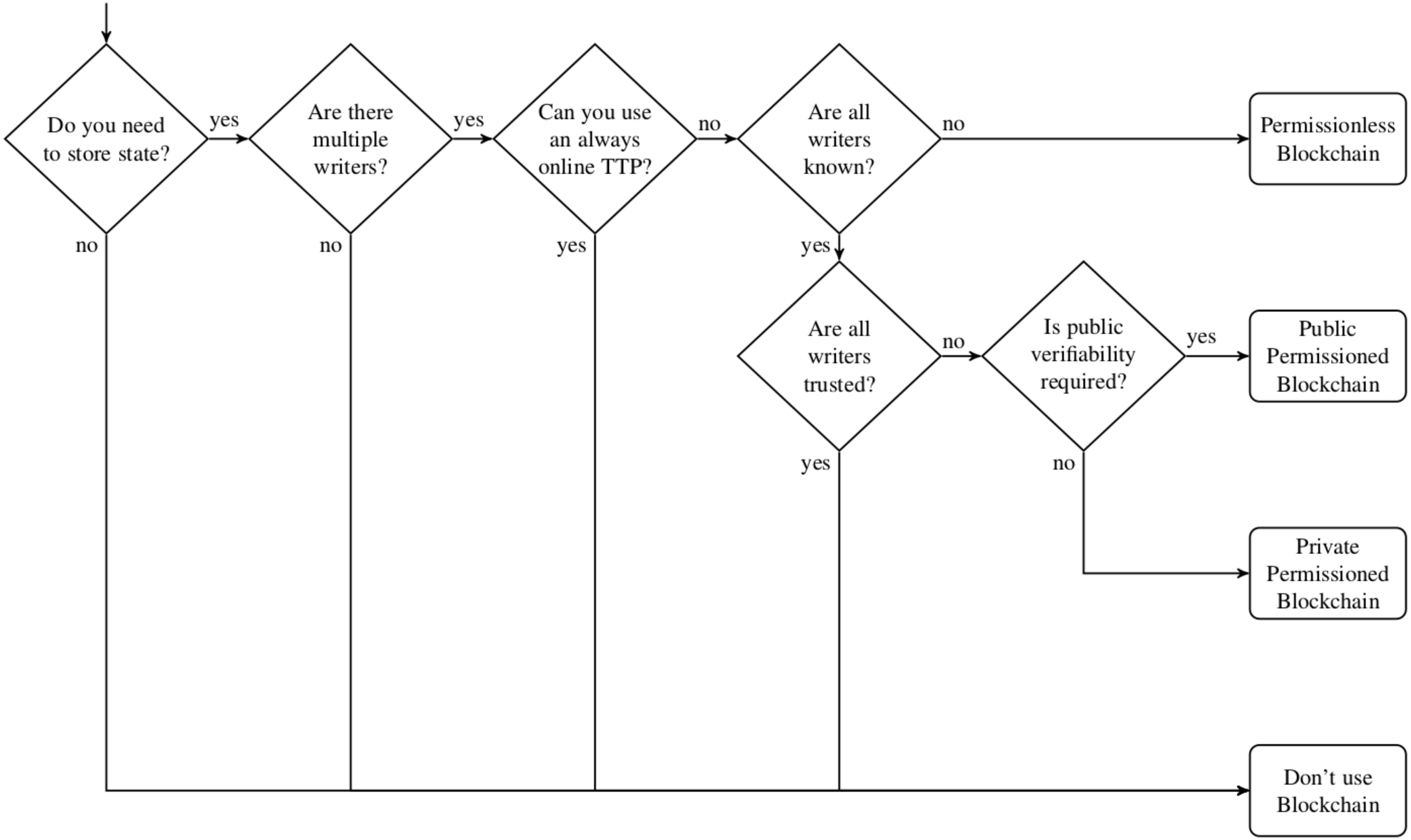
Timings are demonstrative

Use your ingenuity and unique style

- Introduction (~30 seconds)
  - Title
  - Team members and prospective main responsibilities
    - This does NOT mean every member does one thing – collaboration is paramount
    - Presentation / HTML front-end / Solidity development / Javascript development / Architectural design / Token engineering / …
- Presentation of the context (~1 minute)
  - Aim of the DApp
  - Why using the blockchain
    - Is it fit for purpose? Why? What are the advantages?
  - What type of blockchain you would use in production and why (private? Public? Permissioned? Permissionless?)
- Approach (~2 minutes)
  - Expected contracts, tokens, communications in-between, etc.
  - Remarks, e.g., app tokens represent this, ether transactions represent that…
- Early prototype showdown (~2 minutes)
  - Contracts, GUI, logos, etc., or sketchy drawings thereof
  - Software architecture diagrams, or sketchy drawings thereof
- Known issues and limitations (~30 seconds)
- Recap (~30 seconds)
- Slack (~30 seconds)

Looking for inspiration? Watch this brilliant video:
https://youtu.be/Iwpi1Lm6dFo
Also, you may want to check out
my personal guidelines on presentations

Blockchain please?


Do you need to store state?
yes
no
Are there multiple writers?
yes
no
Can you use an always online TTP?
no
yes
Are all writers known?
no
yes
Permissionless Blockchain
Are all writers trusted?
no
yes
Is public verifiability required?
yes
no
Public Permissioned Blockchain
Private Permissioned Blockchain
Don't use Blockchain

Karl Wüst, Arthur Gervais: *Do you need a Blockchain*? IACR Cryptology ePrint Archive 2017: 375 (2017). http://eprint.iacr.org/2017/375

Diagrams

- A UML **concept** diagram (or an analogous model) to describe the structure and content of the **smart contract**(s)

- A UML **component** model (or an analogous diagram) to indicate the back-end and front-end **modules** and their **dependencies**

- UML **collaboration** diagrams (or analogous ones) to illustrate the interactions among those components

- A UML **use case** diagram to illustrate the main usage scenarios of the DApp or tokens

- UML **activity diagrams** (or analogous ones) to illustrate the lifecycle of the tokens

- Happy diagramming!

Structure of the report (use the Sapienza template)

Weights in pages are demonstrative

- Preface (~1 page)
  - Title and 3-sentence presentation of the DApp
  - Team members and main responsibilities
  - Outline of the report
- Background (~5 pages)
  - Blockchain: history, rationale, concepts...
  - Application domain
- Presentation of the context (~3 pages)
  - Aim of the DApp
  - Why using a blockchain, and what type thereof to use in production
- Software architecture (~12 pages)
- Implementation (~6 pages)
  - Like the demo
- Known issues and limitations (~0.5 pages)
- Conclusions (~0.5 pages)
- References (~1 page)