UNIVERSITÀ DEGLI STUDI
DI MILANO

Online Gaming Infrastructures

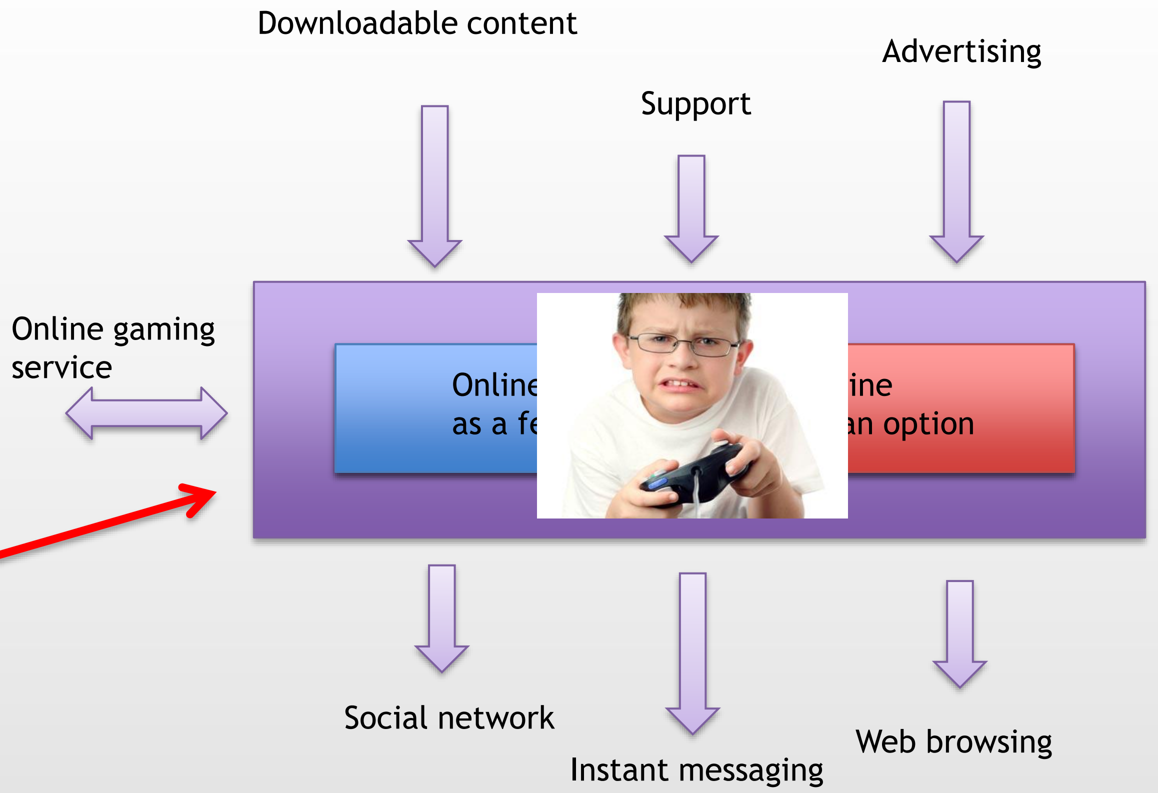
Lesson I03

Dario Maggiorini (dario@di.unimi.it)

Online Game Design - A.A. 2021-2022

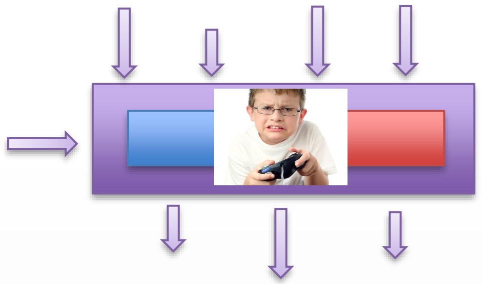
Next Step

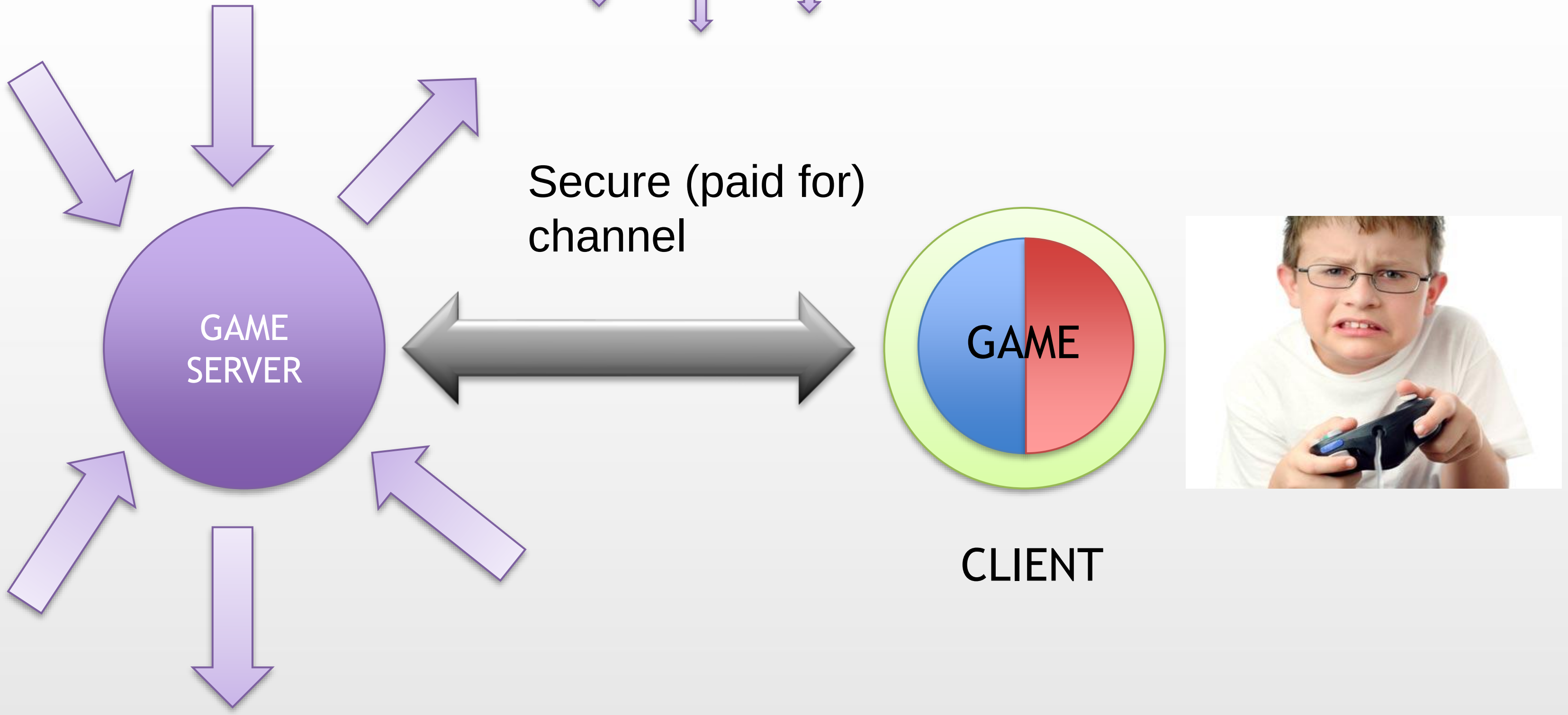
- Now we know why this is important (and complex)
- Time to do *Computer Scientists* work and ... build one



We must go from this ... to a technical infrastructure

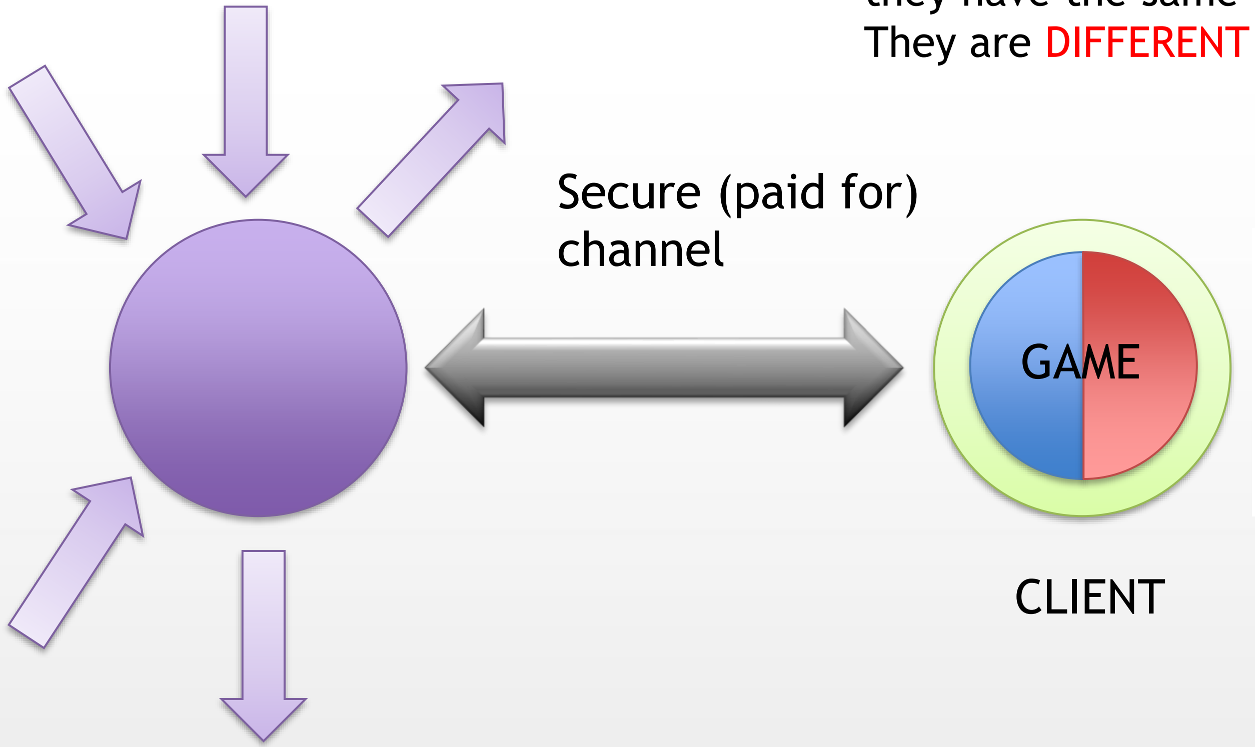
If it is a Service … There is a Server!





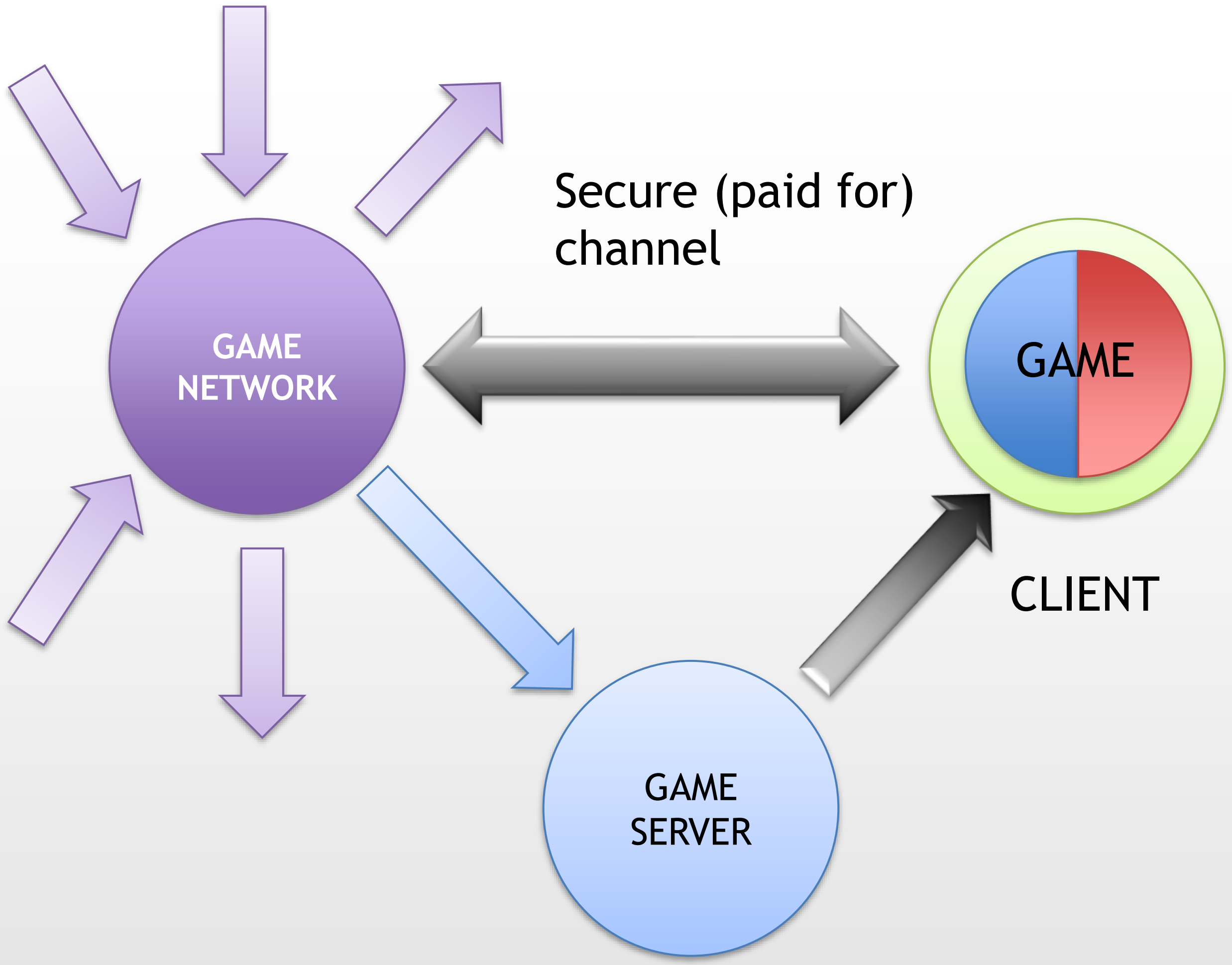
But Life is Never so Easy (in ICT)

Do not get fooled by the fact
they have the same name (e.g., "SONY").
They are DIFFERENT COMPANIES UNDER THE SAME BRAND



The company providing us the added value infrastructure is NOT interested in providing the game.
It simply has another business

But Life is Never so Easy (in ICT)



Relationship Between Game Network and Server

- The game company is a CUSTOMER of the infrastructure company
  - They are going to PAY to connect to the infrastructure
- The game company will request services to the infrastructure
  - Authentication
  - Leaderboards
  - Matchmaking
  - Socials

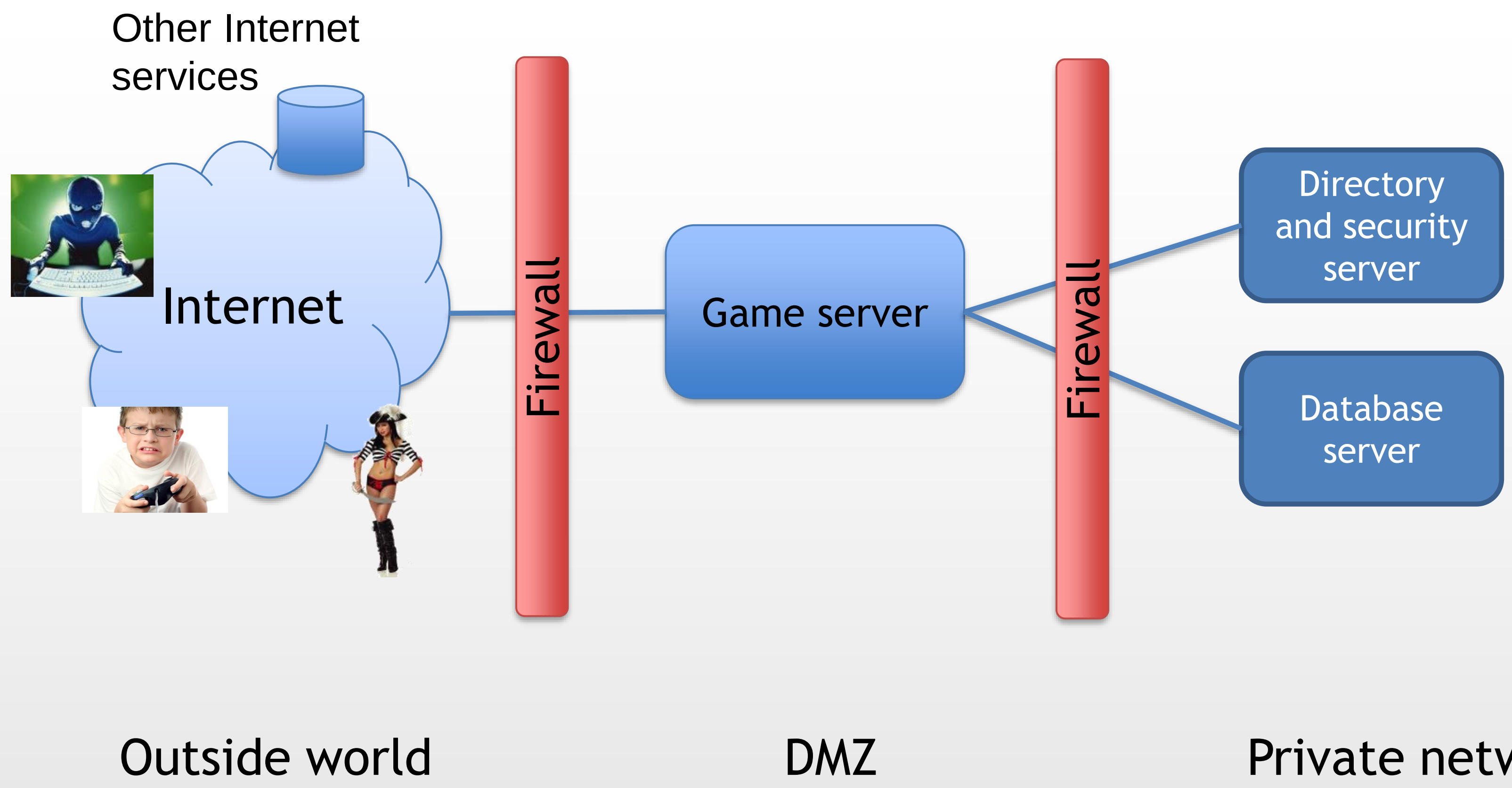
Actually, they are usually forbidden to do that by themselves

Relationship Between Game Network and Server

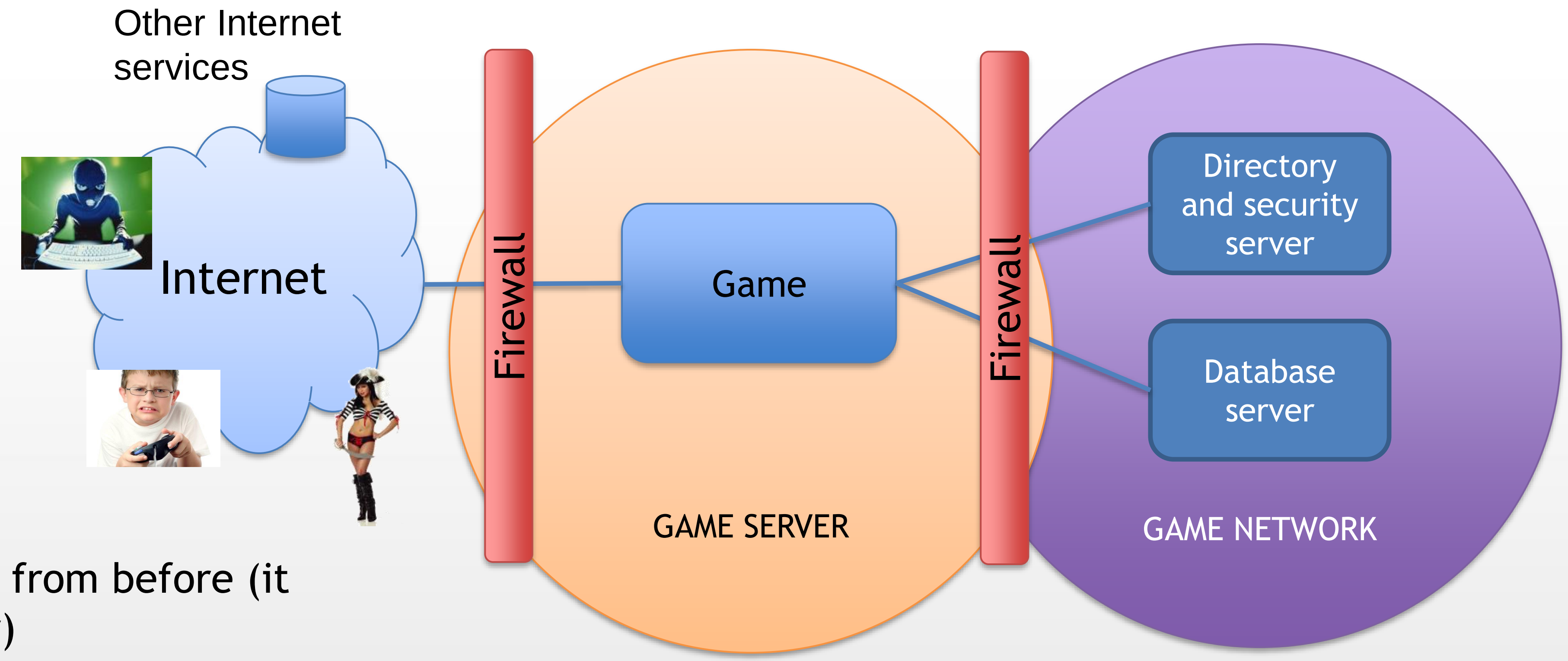
- The game company will deal with the “player” and NOT with the person
  - They will not know your name, just your alias/nick/gamertag
  - But this is extremely good for the game company (DGPR again)

- All game companies connected to the same game network will offer a uniform set of services to their players
  - This will help achieve a smoother player experience between games and increase fidelity (customer retention)

“Just buy a computer and install it!”

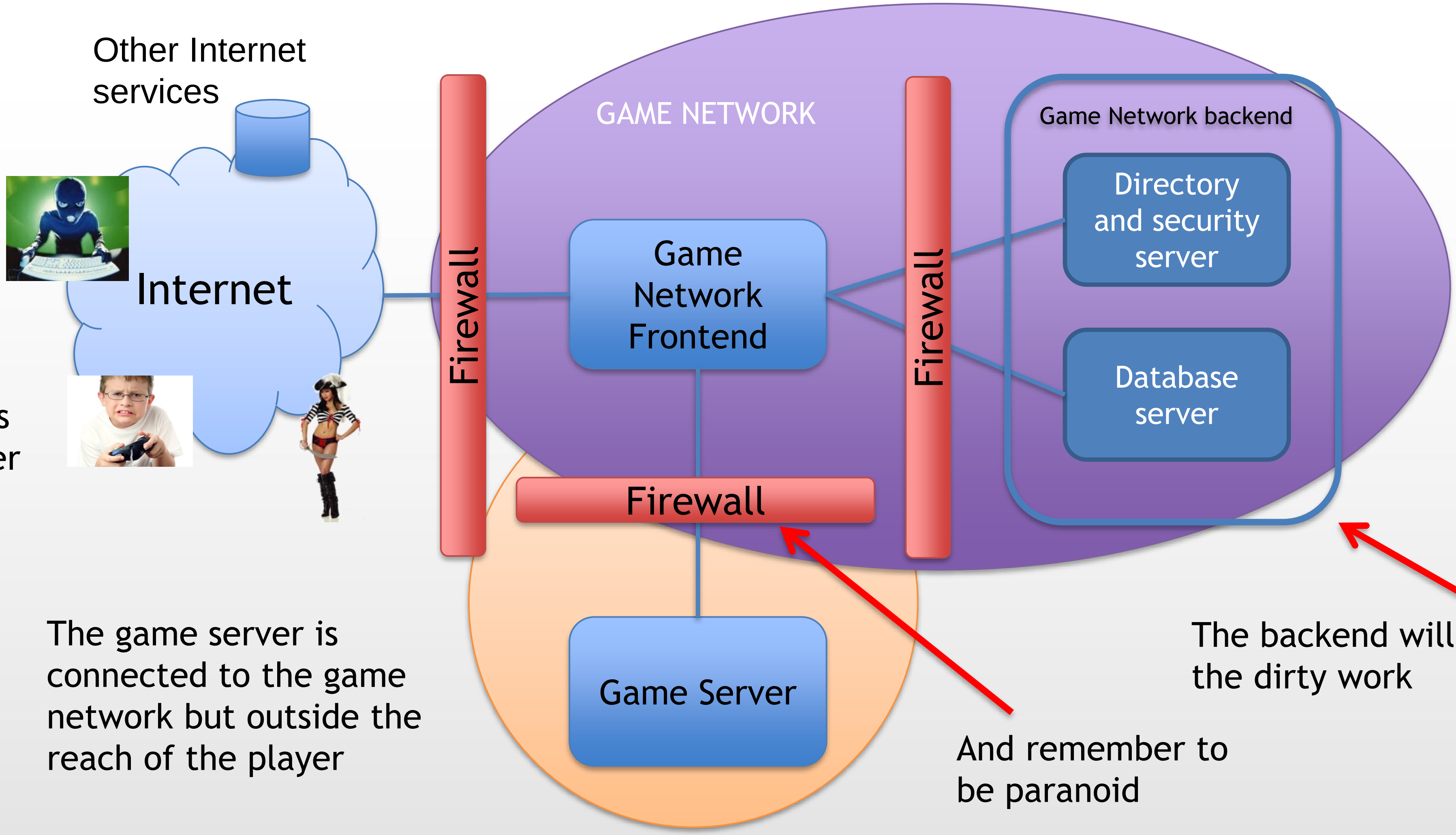


Let’s Put Everything in a Box



This is different from before (it is not triangular)
Let’s just consider this a more “*vintage approach*”

Let’s Put Everything in a Triangular Box



The game network is everything the player needs to know

The game server is connected to the game network but outside the reach of the player

And remember to be paranoid

The backend will do all the dirty work

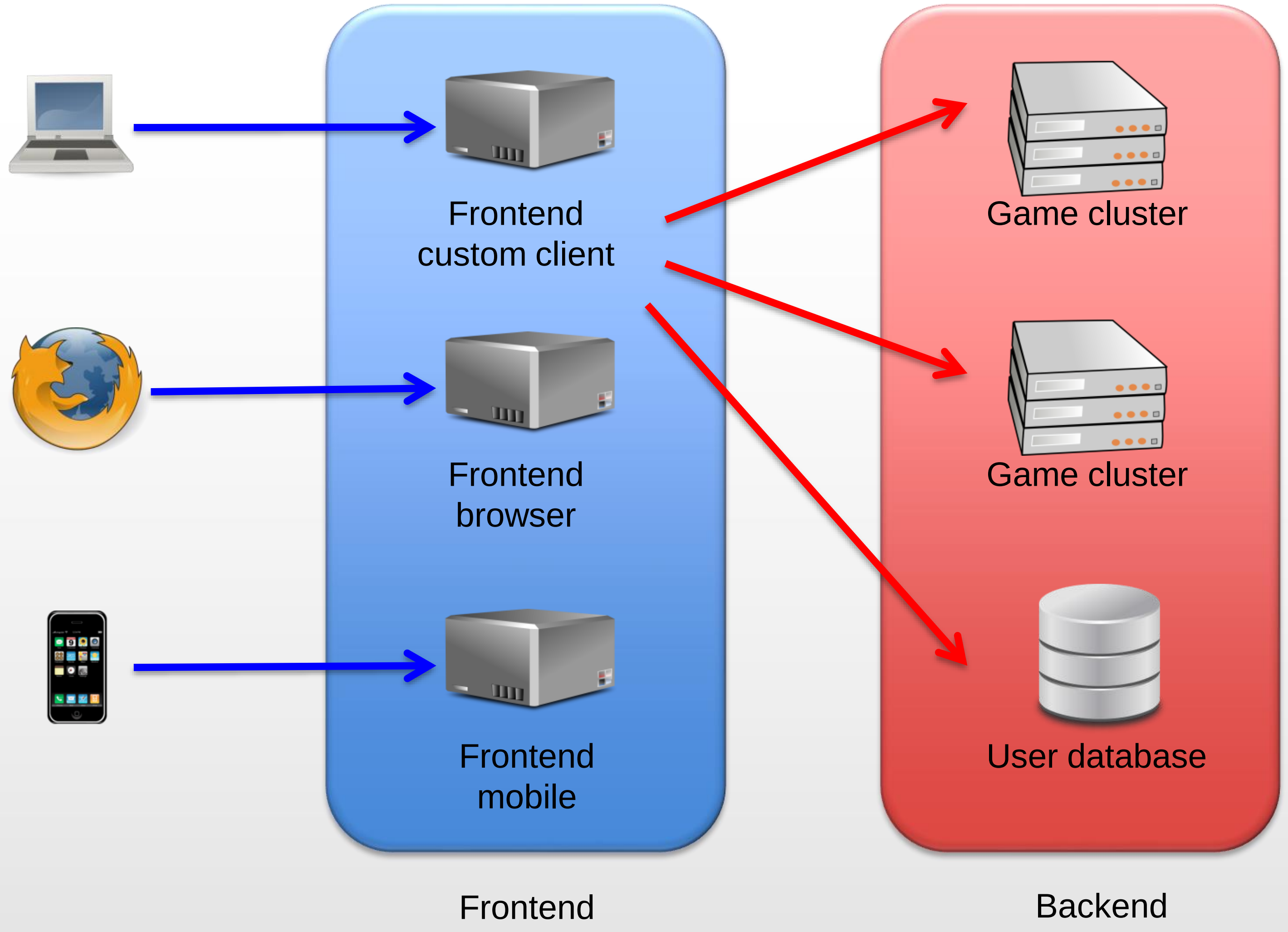
Frontend and Backend

- The two-tier classical model has no direct mapping to modern gaming architectures
  - It is strictly linked to the client-server provisioning model
- Nevertheless, we still talk about frontend and backend as an abstraction

1. Frontend “welcomes” clients, perform initial authentication, and converts messages to a common format understandable from the backend
2. Backend is all the rest without distinction between game server and game network
   - Backoffice services such as authentication and data storage
   - The actual game software

- Beware. We ALWAYS have a multi-layered organization
  - Each backend might have its own backend, like in a matryoshka

High Level Game Network Architecture



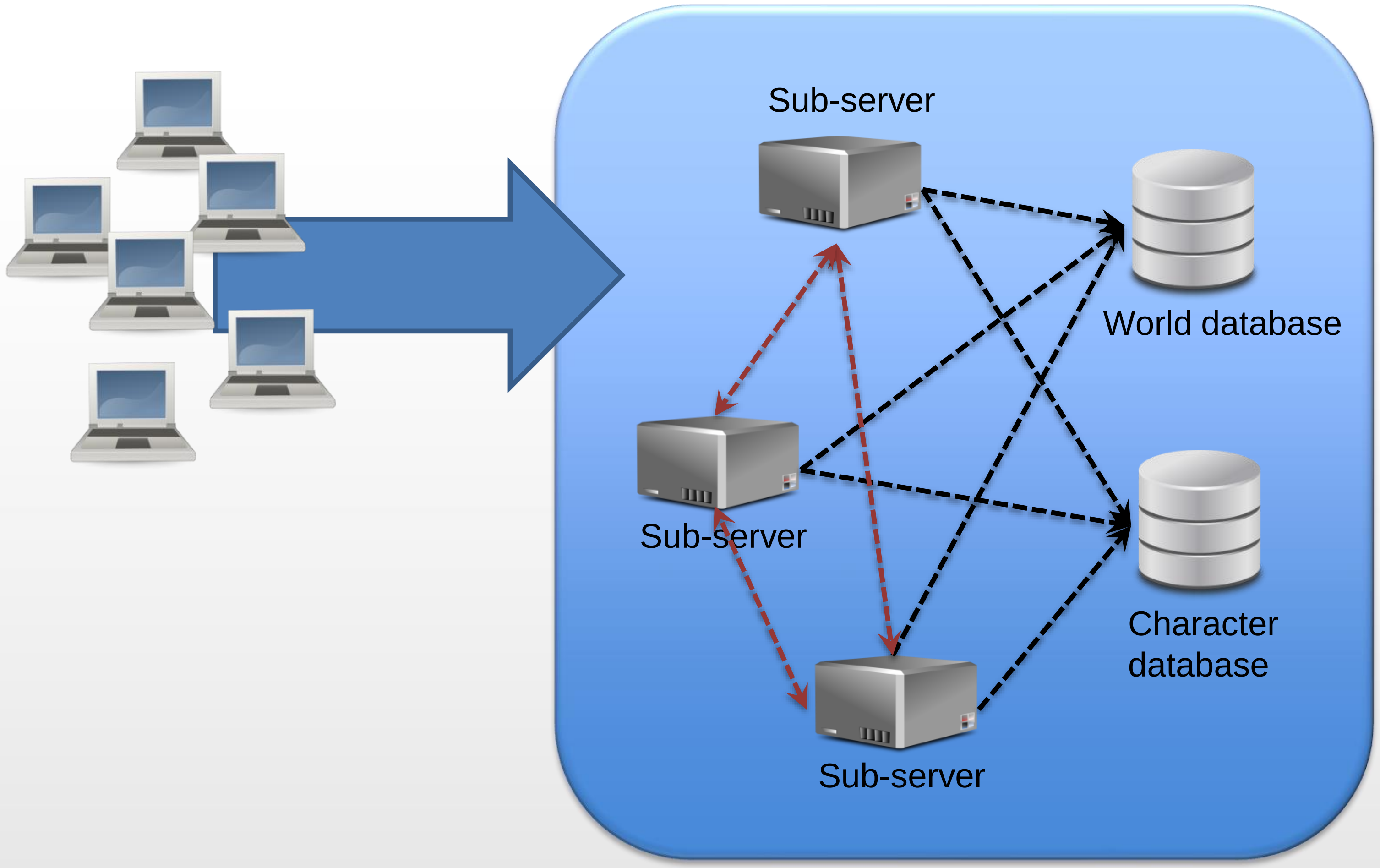
Game Clusters

- The backend may be backing the whole game infrastructure, but it is hosting also systems dedicated to run games
  - These systems fall under the name of *game clusters*
- May be different, independent, incarnations of the same world
  - Also called shards
  - Such as realms in World of Warcraft
- May be completely different worlds
- May be different subsections of the same world
- May be completely different games
- What we call “game server” may be the cooperation of two or more game clusters

Game Clusters Architecture

- Guess what ?
- Also a game cluster is NOT a single machine rather than a combination of smaller ones
- Matryoshka effect again!

Clustered Architecture



Sub-servers

- Are usually independent sections of a game word
  - This way we can increase computational and transmission capacity
- May be taking care of different map fragments
  - Statically o dynamically
- May be taking care of different functions of the game server
- Whatever they are, this is usually some implementation of a clustering technology
  - NOTE: this time, “clustering” is indicating the hardware technology used to implement high availability and fault resilience in datacenters

Cluster Internal Databases

- They are used to stock cluster-specific information about the game
  - Status of the map
  - Connected players
  - Non-relocatable resources
  - ... and whatever is not “backend-wide”

- They are usually divided between environment and players
  - Environment: the backend database will hold maps and blueprint while the cluster database will just keep the differences (deltas) to make that map “special”
  - Player: the backend database will hold authentication and statistic data (about the player) while cluster database will keep avatar information and local resources such as avatars’ skin and inventory

Why clustering?

- Simple: to balance workload!
  - To increase the overall number of players
  - To increase processing power
    - To add detail to the worlds
    - To improve mobs A.I.

Workload

- We may estimate workload in two ways:
  - The number of players per sub-server
    - Each server is assigned a fair share of the total players
    - Big overhead: players mostly move and interact (PvP or PvE)
    - Data synchronization is a HUGE problem
    - The number of players is not a good metric for many reasons. First and foremost the fact that not all the players behave in the same way
  - The level of processing per sub-server
    - Each server is assigned a subsection of the world
    - Each server will take care of interaction and mobs in its area
      - (99% of the cases)
    - Servers talks to each other only for (few) long-range issues
      - (1% of the cases)
    - This solution is a VERY BAD option when you have seasonal or geographical events and all players converge on a single sub-server

Load Balancing (For CPU Workload)

- Fixed load balancing
  - Maps are designed once and for all
  - Easy to implement
  - Will manage a dedicated section in the database with great efficiency
  - Client can pre-fetch environmental data

- Dynamic load balancing
  - Geographic areas may change over time depending on other factors
  - Crossing boundaries may cause troubles
  - Offer better load distribution for seasonal events

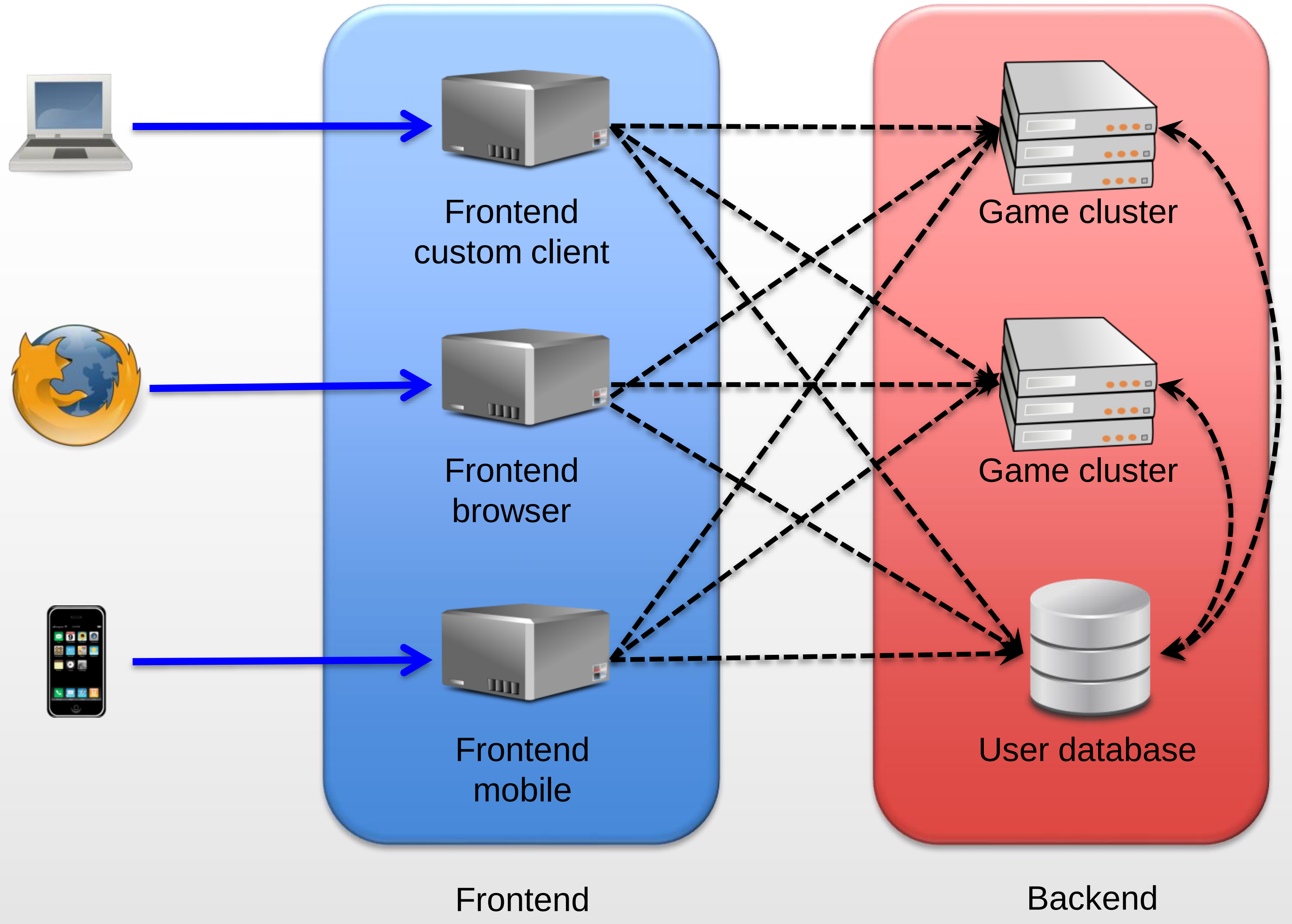
- This is actually a case where technology puts a constraint
  - **Your choice here is going to influence world design and mobs spawning points**

About Databases

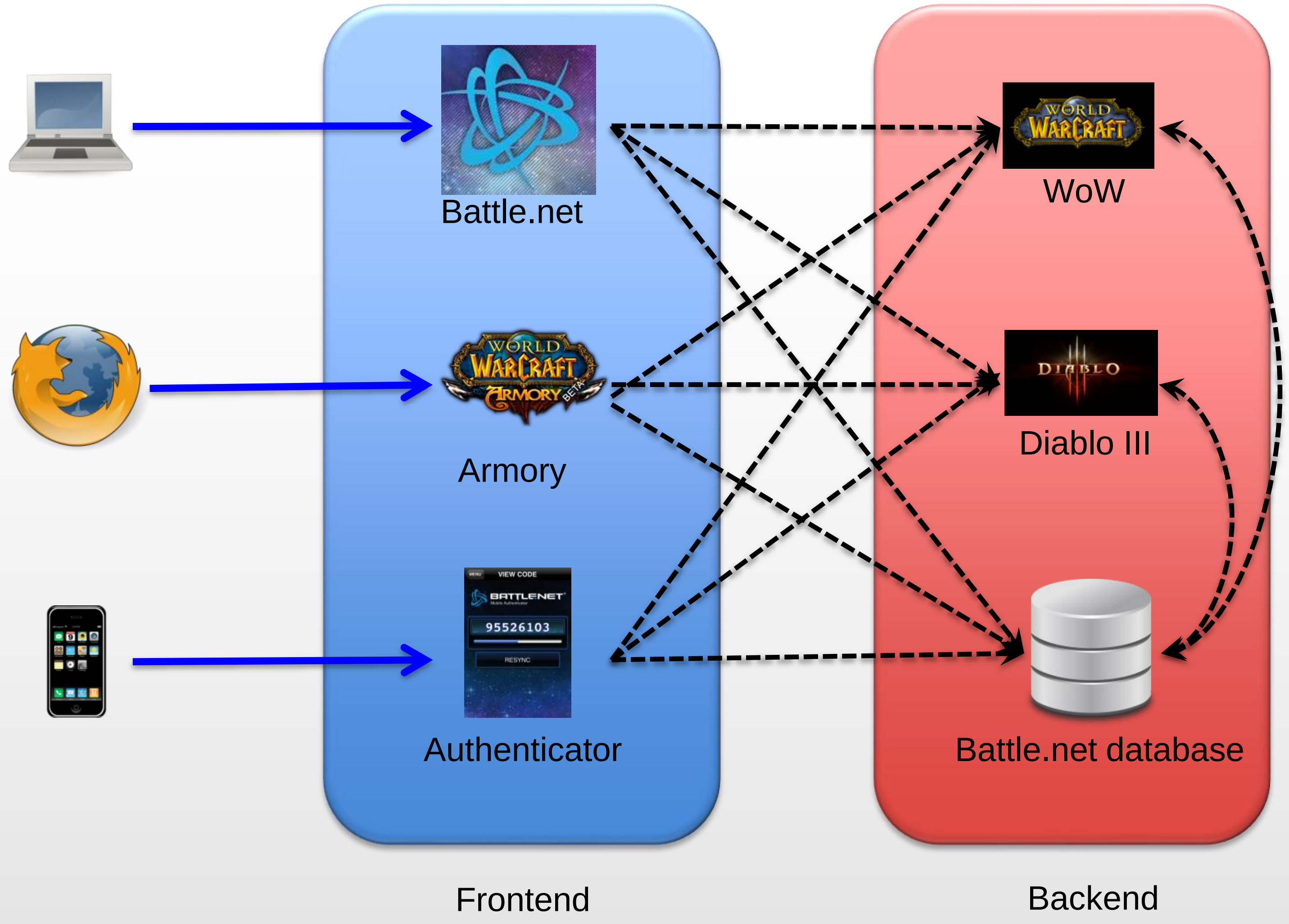
- You need to store a lot of stuff
  - Scripting code
    - Not everything may be hardcoded in the main server
    - What about LUA stored procedures ?
  - Templates
    - Blueprints for objects which may be created in your world
  - Instantiation
    - The current state of your virtual world
  - User data
    - Authentication and accounting (and billing)
  - Character data
    - Stats and permissions
- Planning disk space is crucial!

  … and databases must also be clustered

High Level Game Network Architecture



An Educated Guess About Blizzard



UNIVERSITÀ DEGLI STUDI DI MILANO

High Availability (a.k.a. HA)

- This means fault-tolerant, not fault-proof
- Clusters balance load but may also provide hardware redundancy
  - Active-passive cluster
  - A machine is idle waiting for its sister to die
  - Somehow, 50% of your money is wasted
- Always remember: being idiot-proof is a completely different issue

Business Continuity

- Is a completely different concept from High Availability
- This is about keeping a steady flow of money
- *"to ensure that an organization's critical business functions will either continue to operate despite serious incidents or disasters"*
- Business Continuity = High Availability
  + a recovery strategy
  + a contingency plan

… and sometimes also a Disaster Recovery site!

A Word of Warning: Make or Buy ?

- Do NOT reinvent the wheel!
  - If something is
    - Already there
    - Working
    - Free (or cheap)
  - Make an effort and use it, do not try to always create “better” code
    - Because your code will **never** be better than a existing commercial product (just forget about it!)
  - This implies **LOOKING** for existing solutions for your problems!
    - In the technical design document I will explicitly look for comparisons with existing solutions

References

- An architectural approach to providing online game infrastructures
  by Veronika Megler

  http://www.opensourcetutorials.com/tutorials/Getting-Started/Site-Planning/concentrate-on-the-game/page1.html

- Designing Virtual Worlds
  1st edition, ISBN 9780131018167
  by Richard Bartle
  § 2.1 (Development) and § 2.2 (On Architecture)