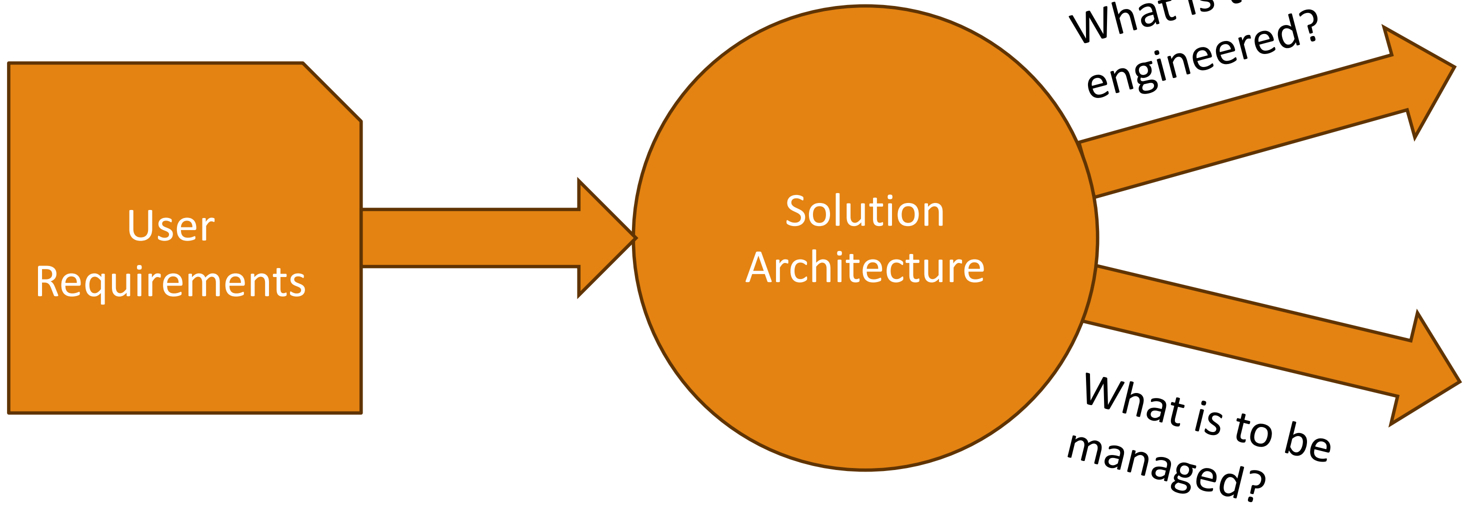
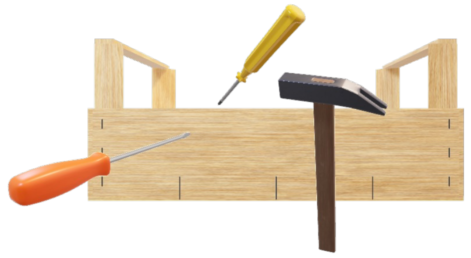
Building Software Systems

Lecture 1.2
Quality Attributes

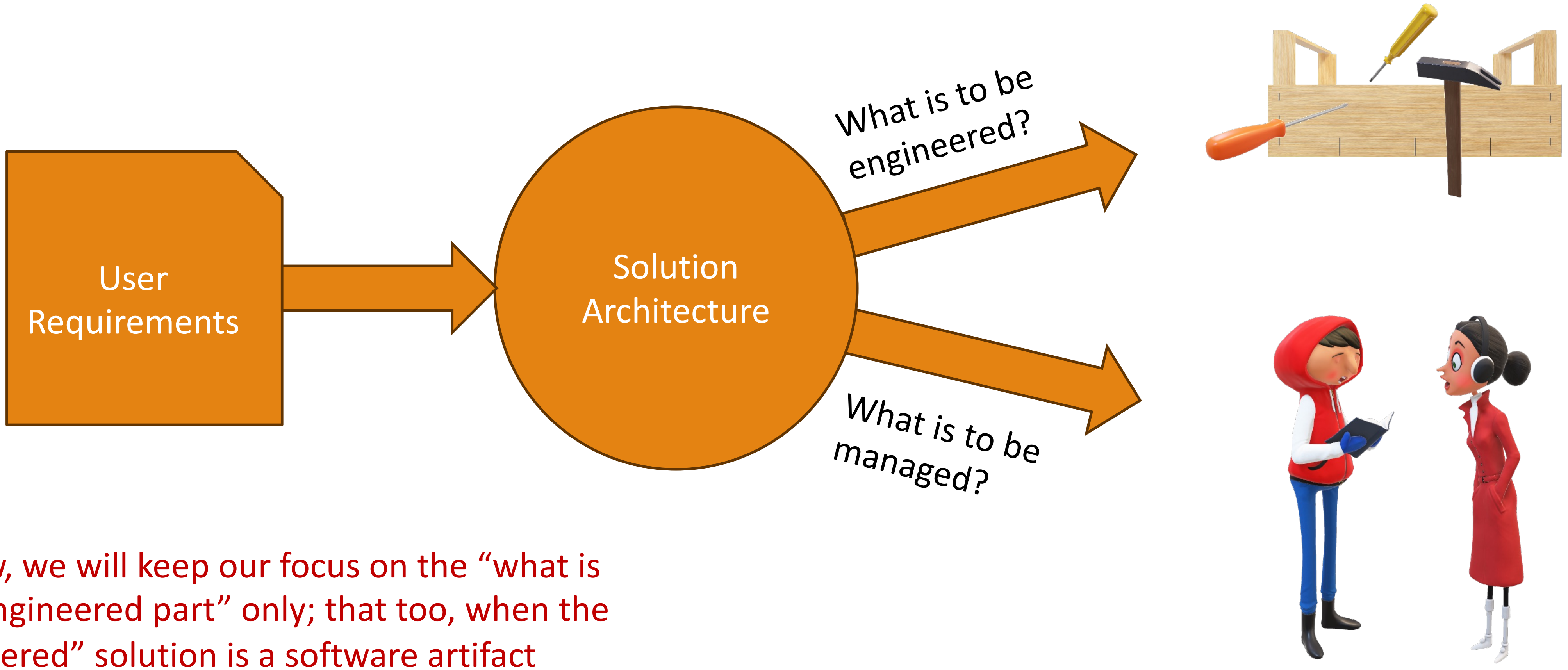
SAURABH SRIVASTAVA
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
IIT (ISM) DHANBAD

Revision – Solution Architecture



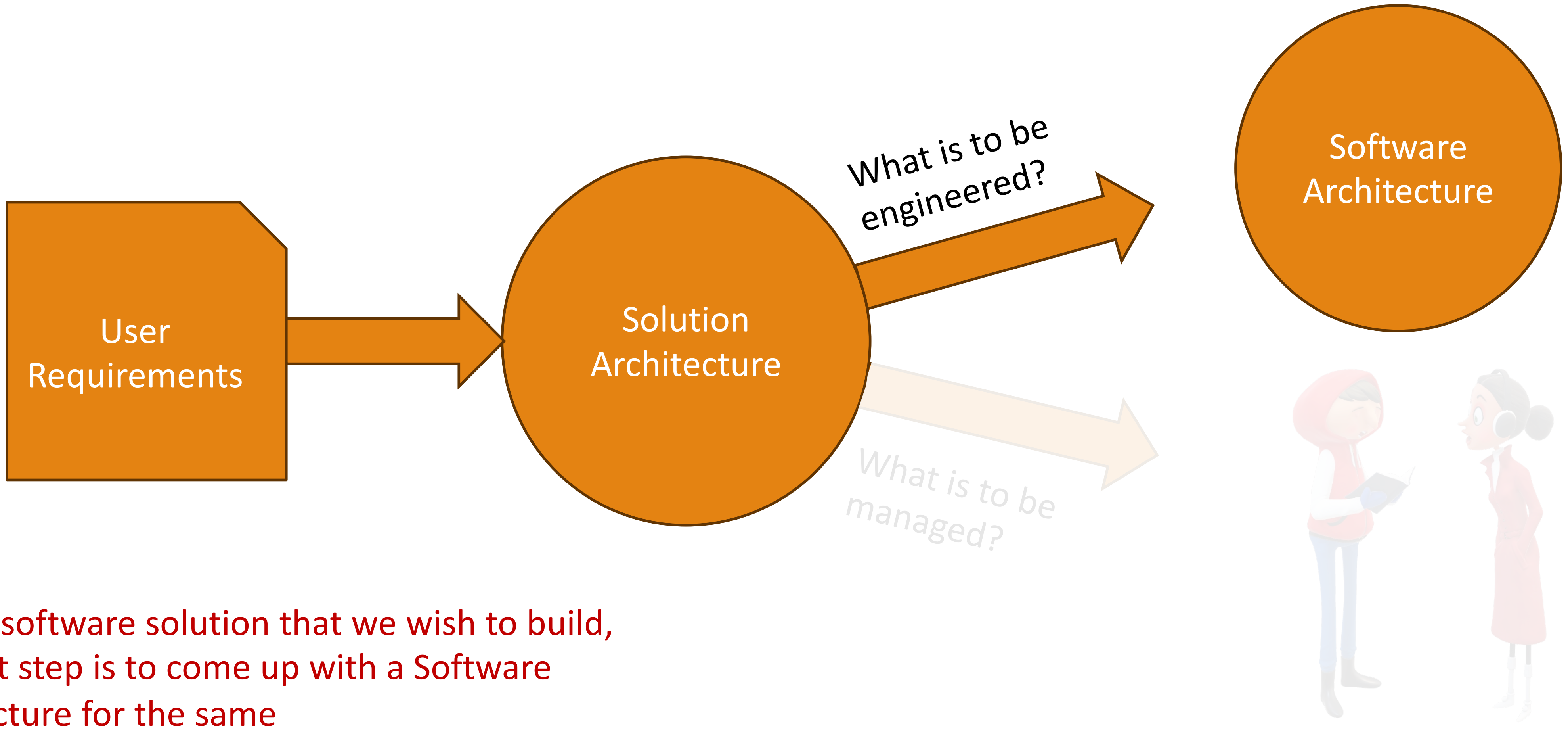


Revision – Solution Architecture



For now, we will keep our focus on the "what is to be engineered part" only; that too, when the "engineered" solution is a software artifact

Next Step – Software Architecture



For the software solution that we wish to build, the next step is to come up with a Software Architecture for the same

What is Software Architecture?

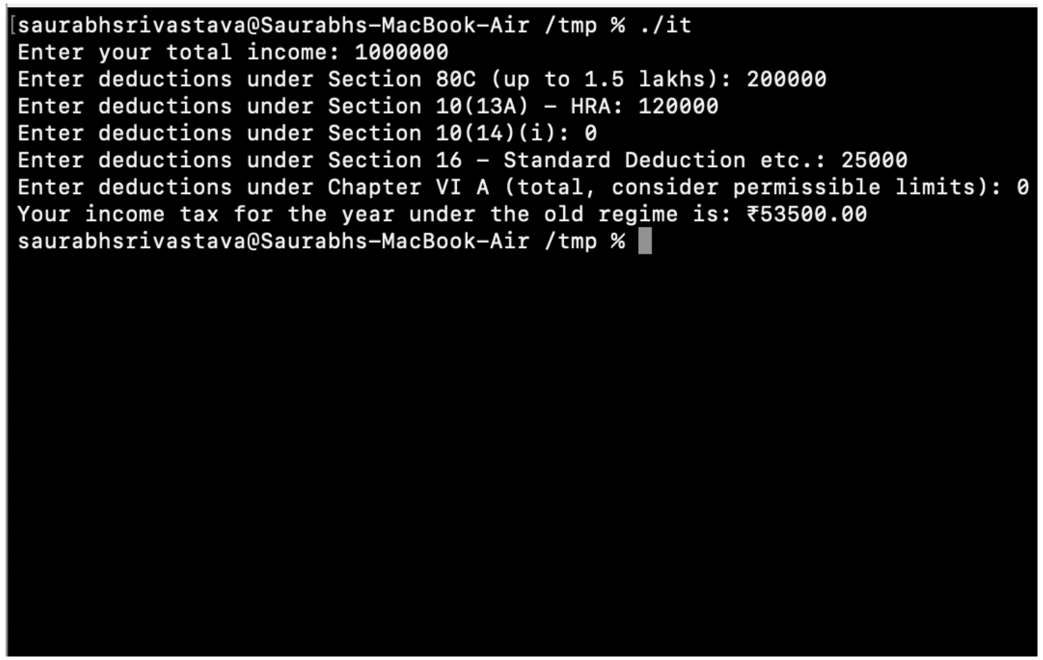
There are actually many definitions for Software Architecture

- There is, in fact a huge collection of them that you can refer to [1]
- For our purpose, let us pick a definition from a book
- *"The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."*
  – [Software Architecture in Practice, Len Bass, Paul Clements and Rick Kazman**,** 3rd Edition, Chapter 1]
- To put it in simple words, you need to look at "smaller structures" that can come together to form …
- … a system that can provide the required functionality with expected quality under the given constraints
- You also need to consider how these smaller structure behave with each other in a cooperative fashion …
- … meaning that you will be required to depict their inter-relationships

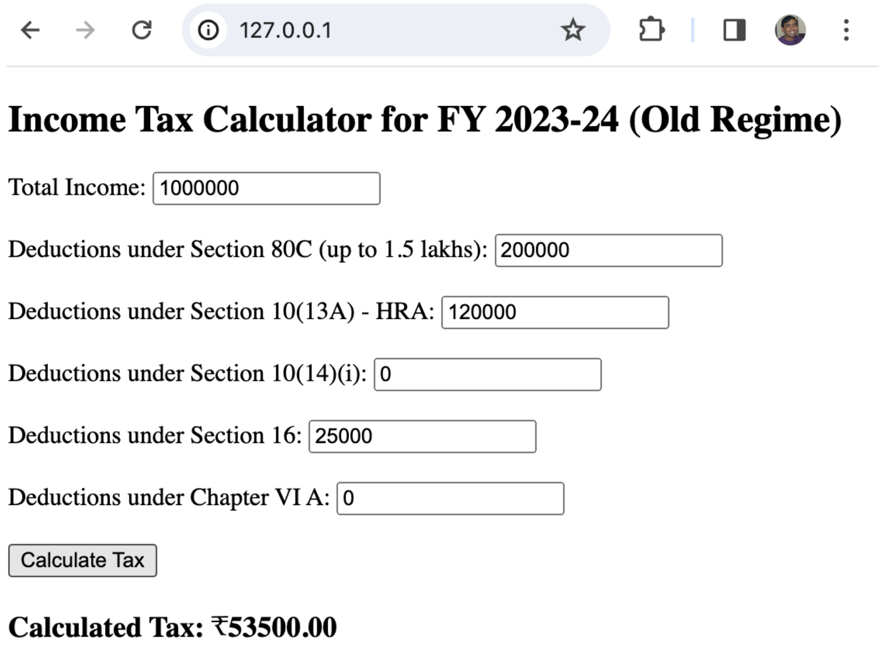
We will come back to Software Architectures in the next Lecture

- But for now, we take a diversion, and talk about the Quality Attributes
- You will understand why we are doing it, when we look at Software Architecture …
- … but for now, let us shift our focus !!

Let us calculate some tax



```
[saurabhsrivastava@Saurabhs-MacBook-Air /tmp % ./it
 Enter your total income: 1000000
 Enter deductions under Section 80C (up to 1.5 lakhs): 200000
 Enter deductions under Section 10(13A) - HRA: 120000
 Enter deductions under Section 10(14)(i): 0
 Enter deductions under Section 16 - Standard Deduction etc.: 25000
 Enter deductions under Chapter VI A (total, consider permissible limits): 0
 Your income tax for the year under the old regime is: ₹53500.00
 saurabhsrivastava@Saurabhs-MacBook-Air /tmp %
```



Income Tax Calculator for FY 2023-24 (Old Regime)

Total Income: 1000000

Deductions under Section 80C (up to 1.5 lakhs): 200000

Deductions under Section 10(13A) - HRA: 120000

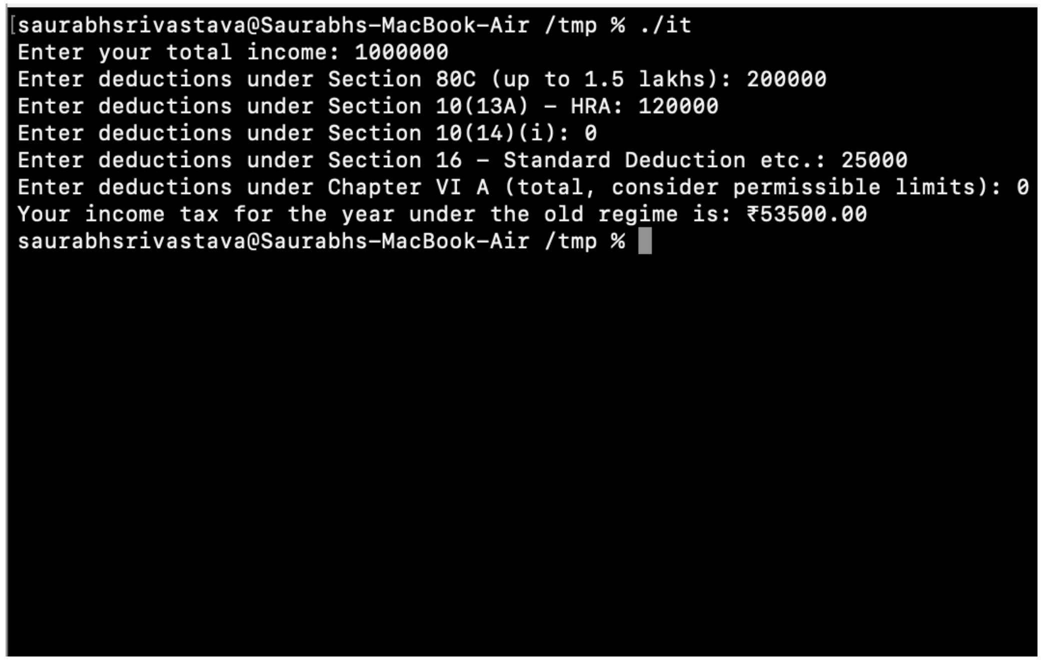
Deductions under Section 10(14)(i): 0

Deductions under Section 16: 25000

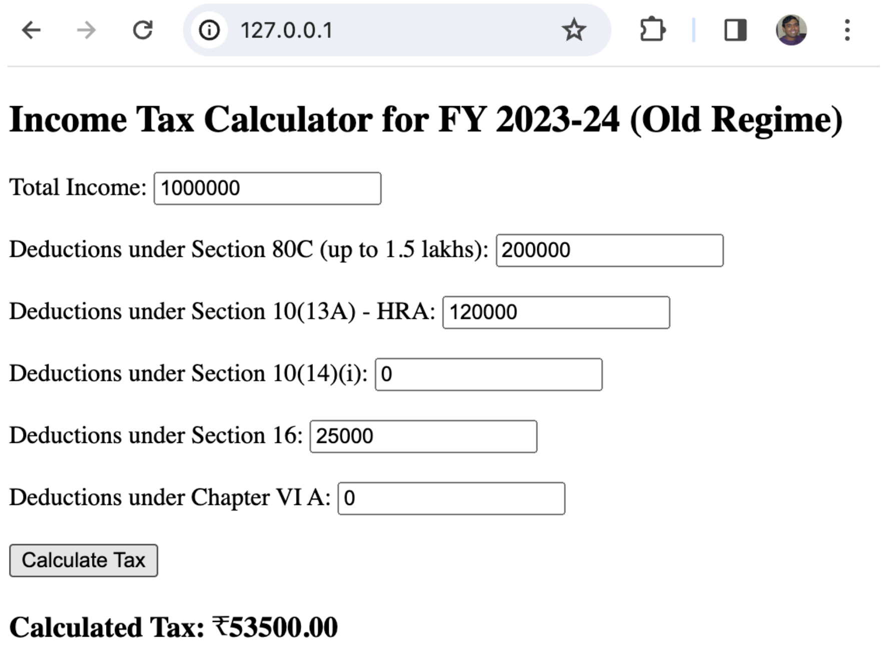
Deductions under Chapter VI A: 0

Calculated Tax: ₹53500.00

Let us calculate some tax



```
saurabhsrivastava@Saurabhs-MacBook-Air /tmp % ./it
Enter your total income: 1000000
Enter deductions under Section 80C (up to 1.5 lakhs): 200000
Enter deductions under Section 10(13A) - HRA: 120000
Enter deductions under Section 10(14)(i): 0
Enter deductions under Section 16 - Standard Deduction etc.: 25000
Enter deductions under Chapter VI A (total, consider permissible limits): 0
Your income tax for the year under the old regime is: ₹53500.00
saurabhsrivastava@Saurabhs-MacBook-Air /tmp %
```



Income Tax Calculator for FY 2023-24 (Old Regime)

Total Income: 1000000

Deductions under Section 80C (up to 1.5 lakhs): 200000

Deductions under Section 10(13A) - HRA: 120000

Deductions under Section 10(14)(i): 0

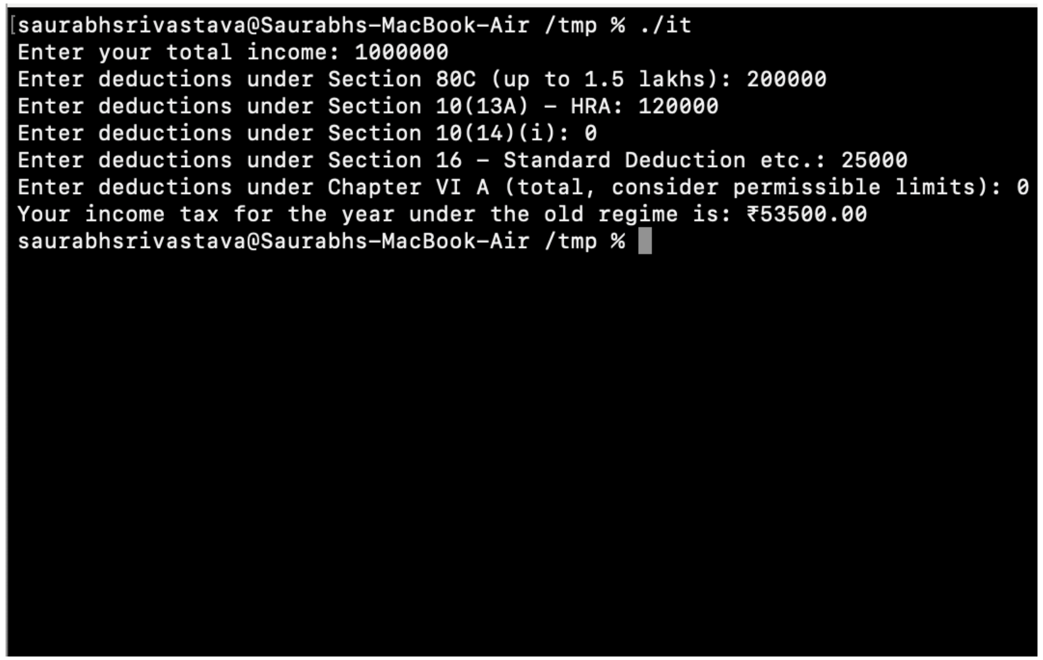
Deductions under Section 16: 25000

Deductions under Chapter VI A: 0

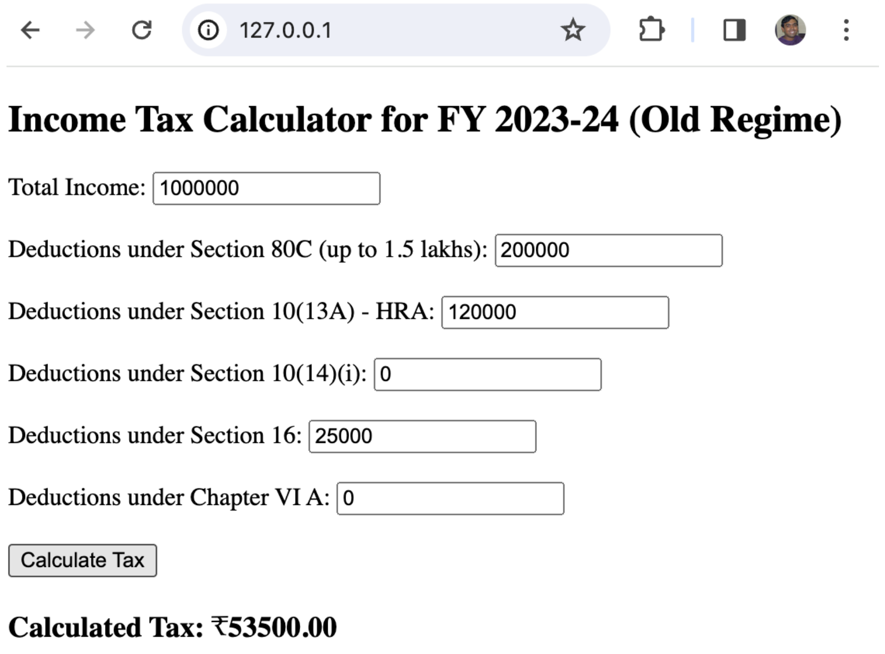
Calculated Tax: ₹53500.00

Which of the two Applications will you rather use?

Let us calculate some tax



```
saurabhsrivastava@Saurabhs-MacBook-Air /tmp % ./it
Enter your total income: 1000000
Enter deductions under Section 80C (up to 1.5 lakhs): 200000
Enter deductions under Section 10(13A) - HRA: 120000
Enter deductions under Section 10(14)(i): 0
Enter deductions under Section 16 - Standard Deduction etc.: 25000
Enter deductions under Chapter VI A (total, consider permissible limits): 0
Your income tax for the year under the old regime is: ₹53500.00
saurabhsrivastava@Saurabhs-MacBook-Air /tmp %
```



Income Tax Calculator for FY 2023-24 (Old Regime)

Total Income: 1000000

Deductions under Section 80C (up to 1.5 lakhs): 200000

Deductions under Section 10(13A) - HRA: 120000

Deductions under Section 10(14)(i): 0

Deductions under Section 16: 25000

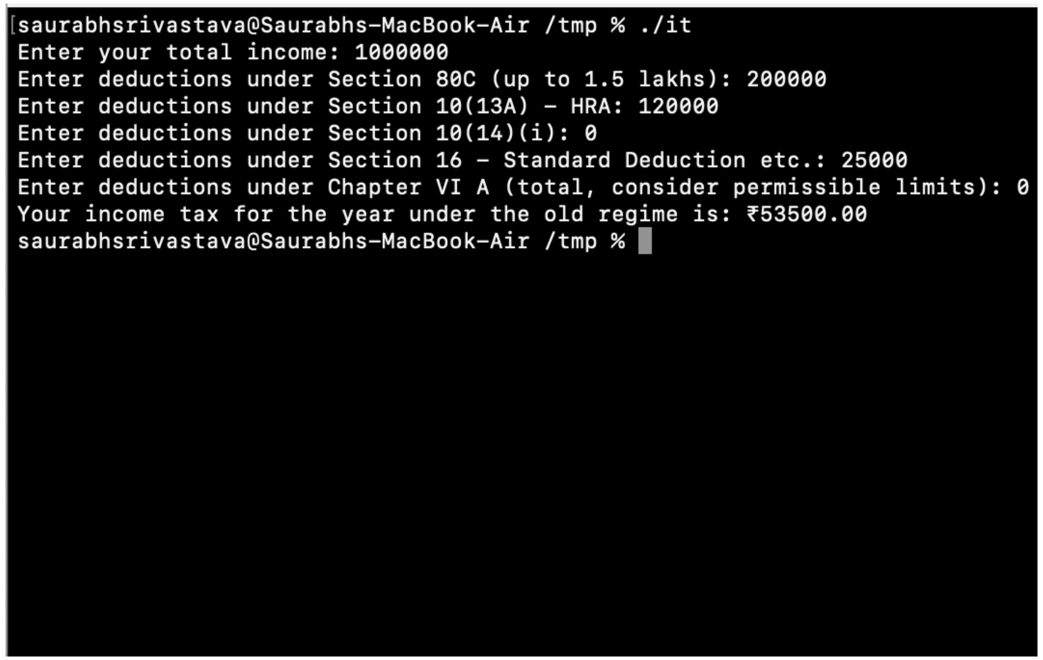
Deductions under Chapter VI A: 0

Calculated Tax: ₹53500.00

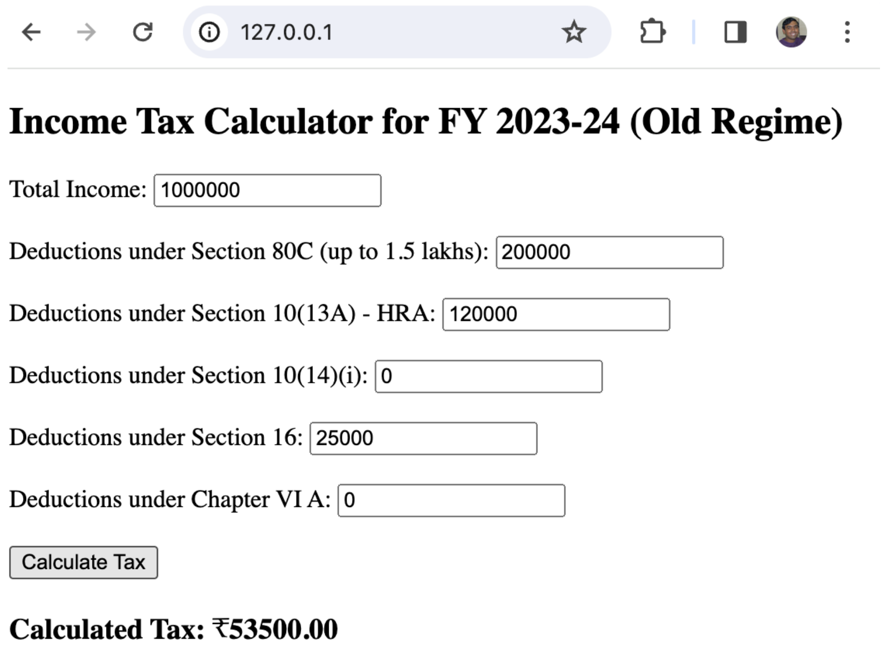
Which of the two Applications will you rather use?

Why?

Let us calculate some tax



```
[saurabhsrivastava@Saurabhs-MacBook-Air /tmp % ./it
Enter your total income: 1000000
Enter deductions under Section 80C (up to 1.5 lakhs): 200000
Enter deductions under Section 10(13A) - HRA: 120000
Enter deductions under Section 10(14)(i): 0
Enter deductions under Section 16 - Standard Deduction etc.: 25000
Enter deductions under Chapter VI A (total, consider permissible limits): 0
Your income tax for the year under the old regime is: ₹53500.00
saurabhsrivastava@Saurabhs-MacBook-Air /tmp %
```



127.0.0.1

Income Tax Calculator for FY 2023-24 (Old Regime)

Total Income: 1000000

Deductions under Section 80C (up to 1.5 lakhs): 200000

Deductions under Section 10(13A) - HRA: 120000

Deductions under Section 10(14)(i): 0

Deductions under Section 16: 25000

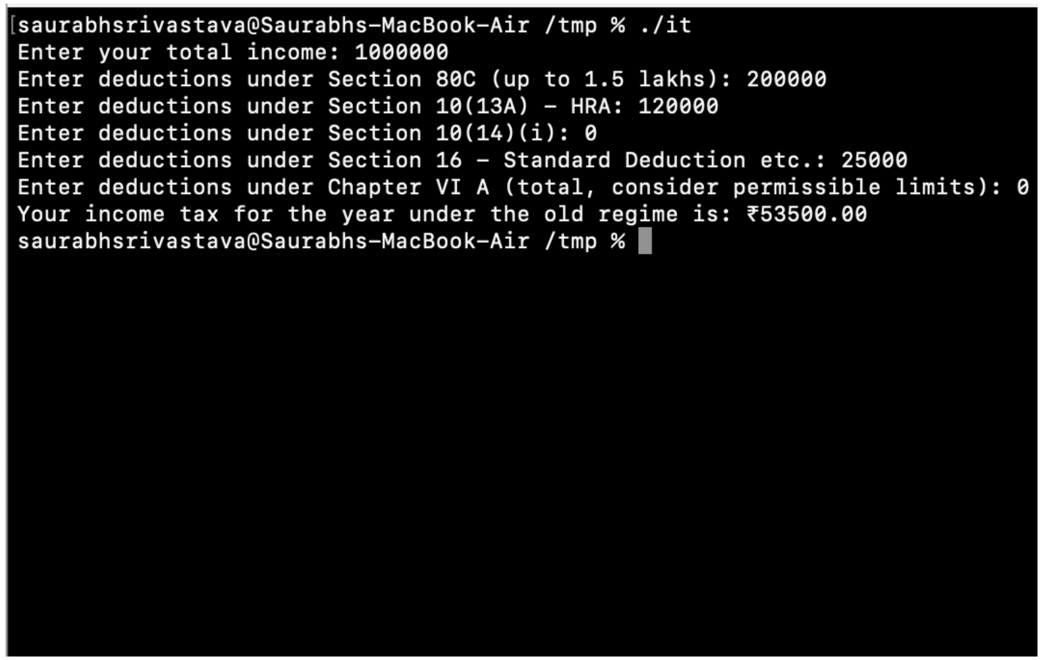
Deductions under Chapter VI A: 0

Calculated Tax: ₹53500.00

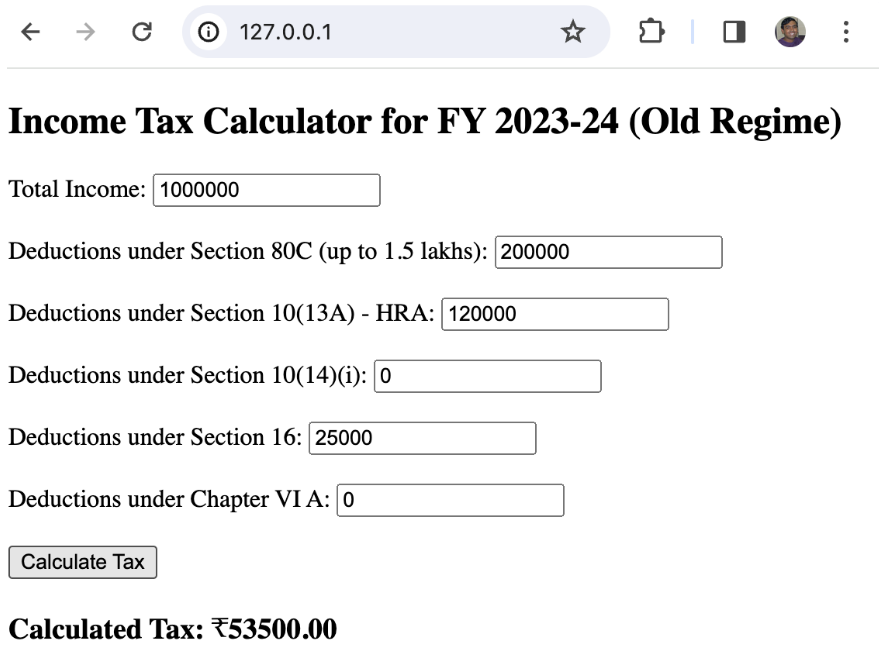
Which of the two Applications will you rather use?

Why? Probably because the one you chose is "better" than the other in "some" aspect

Let us calculate some tax



```
[saurabhsrivastava@Saurabhs-MacBook-Air /tmp % ./it
Enter your total income: 1000000
Enter deductions under Section 80C (up to 1.5 lakhs): 200000
Enter deductions under Section 10(13A) - HRA: 120000
Enter deductions under Section 10(14)(i): 0
Enter deductions under Section 16 - Standard Deduction etc.: 25000
Enter deductions under Chapter VI A (total, consider permissible limits): 0
Your income tax for the year under the old regime is: ₹53500.00
saurabhsrivastava@Saurabhs-MacBook-Air /tmp %
```



127.0.0.1

Income Tax Calculator for FY 2023-24 (Old Regime)

Total Income: 1000000

Deductions under Section 80C (up to 1.5 lakhs): 200000

Deductions under Section 10(13A) - HRA: 120000

Deductions under Section 10(14)(i): 0

Deductions under Section 16: 25000

Deductions under Chapter VI A: 0

Calculated Tax: ₹53500.00

Which of the two Applications will you rather use?

Why? Probably because the one you chose is “better” than the other in “some” aspect – Quality !!

The concept of Quality

Quality is an extremely common word – we use it in context of almost every product we use

- Yet, there is no universally accepted definition for Quality
- Usually, it is easier for people to make a comparison of Quality …
- … for instance, if we have two similar products, they may be able to pick one with “better quality” …
- But formalising the process or attributes to a protocol or specification is usually a hard job !!

Considering this, the following observation (in context of Software products) provides an intuition:

“User satisfaction = compliant product + good quality + delivery within budget and schedule”

Defining Quality Intuitively, IEEE Software, vol. 15, no. 3, pp. 103-104
[**Robert L. Glass**]

This can be intuitively understood as – “Quality is an aspect of User satisfaction”

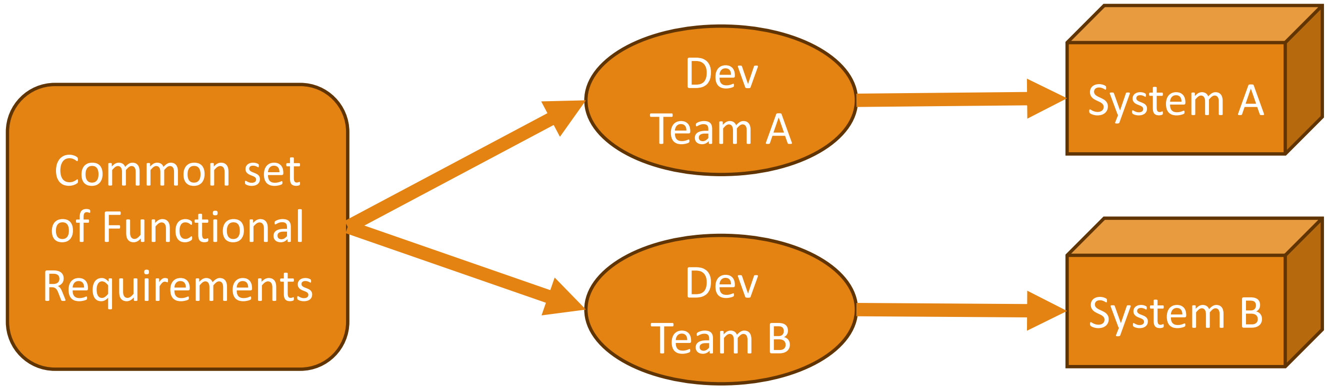
- This hints towards the idea that Quality is linked to user’s experience with the Software Product
- While this is certainly not wrong, it will not be right to consider this as the only aspect of Quality

Defining Quality Attributes

As mentioned before, a crisp definition of Quality is rather difficult

- However, there are some attempts at defining *Quality Attributes* of a system
- *"A quality attribute (QA) is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders."*
  – [Software Architecture in Practice, Len Bass, Paul Clements and Rick Kazman, 3rd Edition, Chapter 4]

One practical representation for Quality Attributes is shown in the diagram below:



System A and System B differ in some Quality Attributes

- An example of the above observation are the two Income Tax Calculation Applications

Quality of Product vs Quality of Process

Quality is not only important for the user, but also for the producer

- For example, a Software Product with "good quality" should require minimal "maintenance overhead"
- It indicates that there are certain aspects of Quality, which are "internal" to the Development Team or Firm

There are many frameworks/standards that attempt to define these "aspects of quality"

- For example, *McCall's Quality Factors*, *ISO 9126*, *ISO 25010* etc.
- Unfortunately, no standard has been accepted for coming up with a de-facto explanation of Software Quality

Still there are at least two categorisations, which are fairly clear in this context

- Quality of the Software Product – it refers to the quality of the built product, which is provided to the user
- Quality of the Software Engineering Process – it refers to the quality of the overall development process
- In this course, we will be focussing on the quality of the software product only …
- … you may know more about the quality of software processes as part of a course in Software Engineering

Common Quality Attributes – Availability

Have you ever heard terms like “four nines availability” or “four nines five availability”?

- Even if you haven’t, its fine !!

These terms refer to the Availability guarantee that service providers guarantee to you

- Availability is measured in terms of the amount of time, a service is available, when it is accessed
- A service is said to be “not available” or “down” if it is not accessible to the users
- The average period for which a system is down in a given amount of time is called its average *Downtime*
- Thus, Availability of a system or a service is inversely proportional to its average Downtime

Availability is usually measured in terms of the “percentage of time a system is not down”

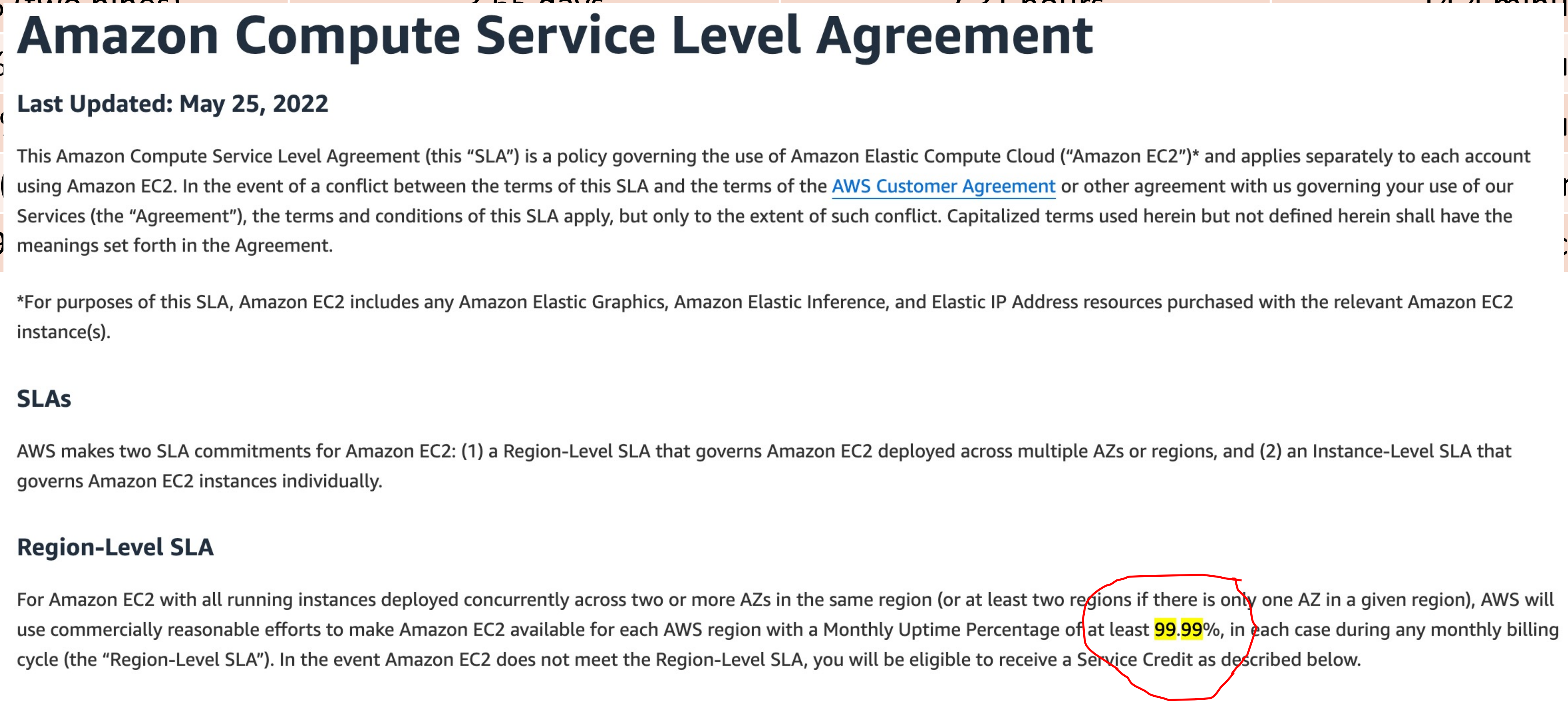
- A “100% Availability” guarantee means that the system or the service has “0” average downtime
- While it may seem tempting to guarantee such a scenario, practically it is usually not a good idea to do so
- Usually, service providers provide guarantees in the range of “99% and above”
- The actual guarantee that they provide gave popularity to terms like “four nines” and “four nine fives”
- These guarantees are provided as part of a Service Level Agreement or SLA between you and the platform

Availability %	Average Downtime per year	Average Downtime per month	Average Downtime per day
99% (two nines)	3.65 days	7.31 hours	14.4 minutes
99.9% (three nines)	8.77 hours	43.83 minutes	1.44 minutes
99.99% (four nines)	52.6 minutes	4.38 minutes	8.64 minutes
99.995 (four nines five)	26.3 minutes	2.19 minutes	4.32 seconds
99.999 (five nines)	5.26 minutes	26.3 minutes	864 milliseconds

Common Availability SLA values

Source: [High availability - Wikipedia]

Availability %	Average Downtime per year	Average Downtime per month	Average Downtime per day
99% (two nines)	3.65 days	7.21 hours	14.4 minutes
99.9%			utes
99.99			utes
99.995 (			nds
99.99			conds



Amazon Compute Service Level Agreement

Last Updated: May 25, 2022

This Amazon Compute Service Level Agreement (this "SLA") is a policy governing the use of Amazon Elastic Compute Cloud ("Amazon EC2")* and applies separately to each account using Amazon EC2. In the event of a conflict between the terms of this SLA and the terms of the AWS Customer Agreement or other agreement with us governing your use of our Services (the "Agreement"), the terms and conditions of this SLA apply, but only to the extent of such conflict. Capitalized terms used herein but not defined herein shall have the meanings set forth in the Agreement.

*For purposes of this SLA, Amazon EC2 includes any Amazon Elastic Graphics, Amazon Elastic Inference, and Elastic IP Address resources purchased with the relevant Amazon EC2 instance(s).

SLAs

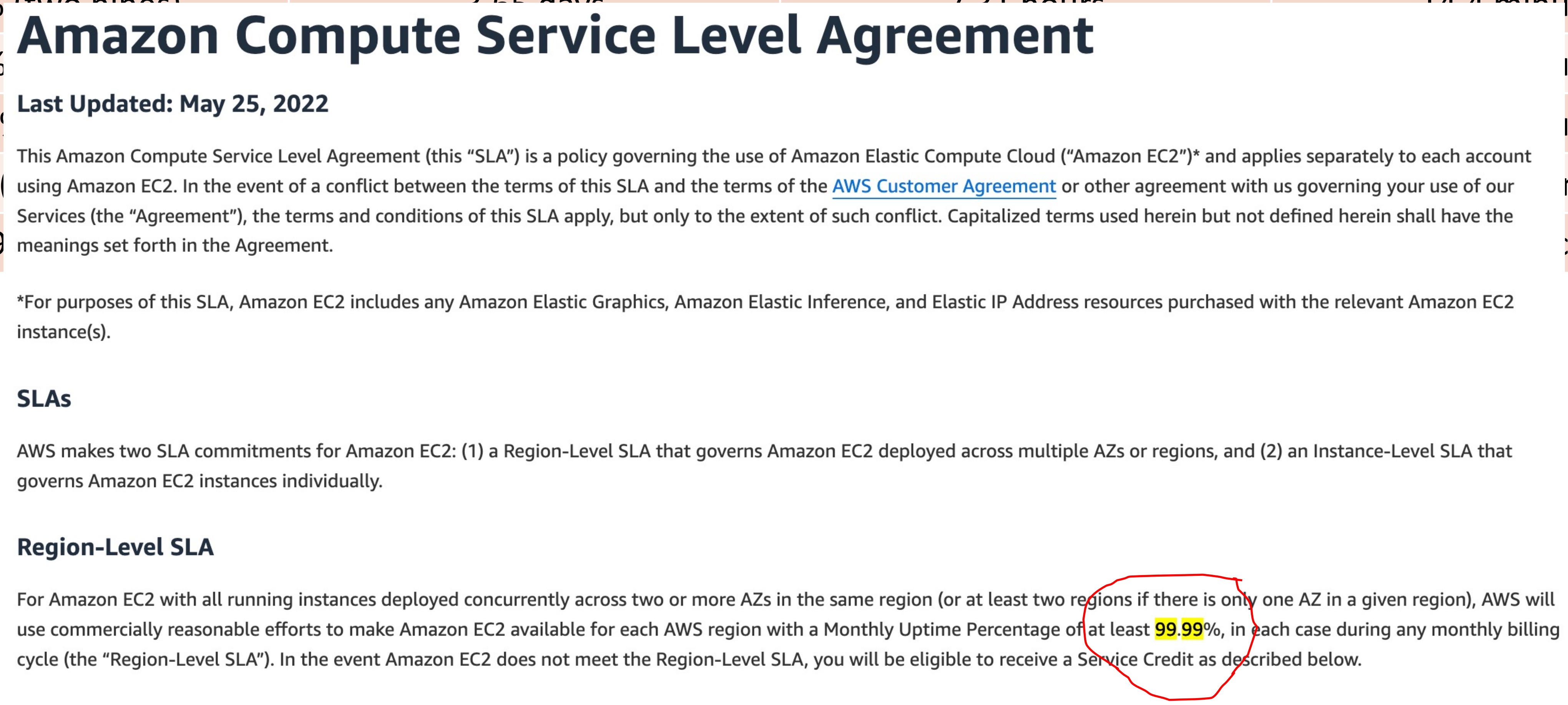
AWS makes two SLA commitments for Amazon EC2: (1) a Region-Level SLA that governs Amazon EC2 deployed across multiple AZs or regions, and (2) an Instance-Level SLA that governs Amazon EC2 instances individually.

Region-Level SLA

For Amazon EC2 with all running instances deployed concurrently across two or more AZs in the same region (or at least two regions if there is only one AZ in a given region), AWS will use commercially reasonable efforts to make Amazon EC2 available for each AWS region with a Monthly Uptime Percentage of at least 99.99%, in each case during any monthly billing cycle (the "Region-Level SLA"). In the event Amazon EC2 does not meet the Region-Level SLA, you will be eligible to receive a Service Credit as described below.

Source: Amazon Compute Service Level Agreement

Availability %	Average Downtime per year	Average Downtime per month	Average Downtime per day
99% (two nines)	3.65 days	7.31 hours	14.4 minutes
99.9%			tes
99.99			tes
99.995 (			nds
99.99			conds



Amazon Compute Service Level Agreement

Last Updated: May 25, 2022

This Amazon Compute Service Level Agreement (this "SLA") is a policy governing the use of Amazon Elastic Compute Cloud ("Amazon EC2")* and applies separately to each account using Amazon EC2. In the event of a conflict between the terms of this SLA and the terms of the AWS Customer Agreement or other agreement with us governing your use of our Services (the "Agreement"), the terms and conditions of this SLA apply, but only to the extent of such conflict. Capitalized terms used herein but not defined herein shall have the meanings set forth in the Agreement.

*For purposes of this SLA, Amazon EC2 includes any Amazon Elastic Graphics, Amazon Elastic Inference, and Elastic IP Address resources purchased with the relevant Amazon EC2 instance(s).

SLAs

AWS makes two SLA commitments for Amazon EC2: (1) a Region-Level SLA that governs Amazon EC2 deployed across multiple AZs or regions, and (2) an Instance-Level SLA that governs Amazon EC2 instances individually.

Region-Level SLA

For Amazon EC2 with all running instances deployed concurrently across two or more AZs in the same region (or at least two regions if there is only one AZ in a given region), AWS will use commercially reasonable efforts to make Amazon EC2 available for each AWS region with a Monthly Uptime Percentage of at least 99.99%, in each case during any monthly billing cycle (the "Region-Level SLA"). In the event Amazon EC2 does not meet the Region-Level SLA, you will be eligible to receive a Service Credit as described below.

This is an example of "four nines" Availability !!

Source: Amazon Compute Service Level Agreement

Common Quality Attributes – Performance

Performance of an Application is usually measured by its “responsiveness”

- This in turn, is usually measured in terms of the *Response Time* of the application …
- … when put under “considerable” load
- In general, a Large-scale application’s response time must not increase beyond acceptable levels …
- … especially till the load of the system is within planned limits
- A term that is often used to refer to systems that follow these properties is *Scalable Systems*

There are usually two ways to achieve the scalability property

- *Vertical Scaling* refers to using “bigger and more powerful” hardware for deploying the system
- While its usually an easy way out, vertical scaling has its limitations
- *Horizontal Scaling* involves using “multiple hardware” components for deploying the system
- Here, the individual components are of relatively modest configurations
- It means more load can be handled by adding more machines (instead of more RAM/cores in a single machine)
- While harder to achieve, modern-day Large-scale applications usually rely on horizontal scaling

Common Quality Attributes – Reliability

Reliability refers to the aspect of a system to be able to function in an error-free fashion

- It is an attribute which is ideally not measured directly, but possibly, in terms of other attributes
- For example, a system with low Availability, can almost always be called "less reliable" as well

For certain systems, it may be too challenging to expect no errors at all

- The best example is our homegrown ERP – MIS – the requirements change so often, that errors are unavoidable
- For such systems, an important aspect that contributes to reliability is "recoverability"
- For example, if an instance of a particular service dies, the system should respawn it and put it to work
- Automated Container Management environments like *Kubernetes* can be configured to do so

Another aspect that can aid Reliability of a system, is its handling of a failure

- The reliability of a "highly-modular" system is usually better than a monolithic one
- This is because even in case of failures, "some part of the system" may still be usable
- **In fact, in this course, we will stress that building a "failure-proof" system is rather wishful …**
- **… but building a system where there is a well-defined action plan for every failure is actually more practical**

Common Quality Attributes – Security

How can you call a system more “secure”?

- In modern world, data is a valuable commodity – it is bought and sold both legally as well as in black market
- One aspect of a secure system, thus, is to have safeguards for the user’s “privacy” …
- … i.e., the user is aware of the data she shares with the system, as well as have a control over its usage

Data – at rest or in motion – needs to be protected from unauthorised access

- Data at rest refers to stored data, e.g., in Databases and Files
- Data in motion refers to transmitted data, e.g., over HTTP/HTTPS

Another important aspect of Security is Authentication and Non-repudiation

- Authentication essentially means that the system should only let *you* in and not someone *impersonating you*
- Non-repudiation means if you performed some actions, you should not be able to deny that you did them

Security is often a neglected Quality Attribute at the beginning of a project

- By the time its relevance is realised, it may already be too late to change the system

Some other Quality Attributes

Before we move further, an important sidenote

- The Quality Requirement for one project, may be a Functional Requirement for another
- For example, for an Anti-virus Solution, Security may have to be considered as a functional attribute
- Similarly, for a real-time system, High-Performance is an utmost necessity, and is not “optional”

Some other Quality Attributes that you should know about include

- **Modifiability:** The ability to change the system effectively in case of changing requirements
- **Portability:** The ability to create different versions of the system, to be used over different environments
- **Interoperability:** The ability of the system to operate with other systems in the environment

Another important Quality Attribute is **Usability** – the ease with which the system could be used

- Usability itself is multi-faceted, e.g., operability, accessibility, learnability etc.

Usually, for every software project, only a subset of the Quality Attributes are “achievable”

- Because there are often trade-offs between them – we will see some examples of these in the next Lecture

In a nutshell …

After the Solution Architecture, we know the scope of software development

- For this development, we need to come up with an architecture
- We will discuss this part in the next lecture

Quality Attributes are actually “implicit” requirements from a software product

- They are often not mentioned explicitly, but are very important for the system
- In fact, they are the ones who shape the architecture of the system, as we will see in the next lecture

Homework

Skim through the ISO 25010 standard

- You may refer this link: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
- While it may not align with our class discussions, it is not a bad reference source

References

1. Carnegei Mellon University, Software Engineering Institute. What is your definition of software architecture? https://insights.sei.cmu.edu/documents/2544/2010_010_001_513810.pdf . (Accessed on 09/01/2024).