CGI

Estimation of a Micro Services based Estimation application

Bhawna Thakur

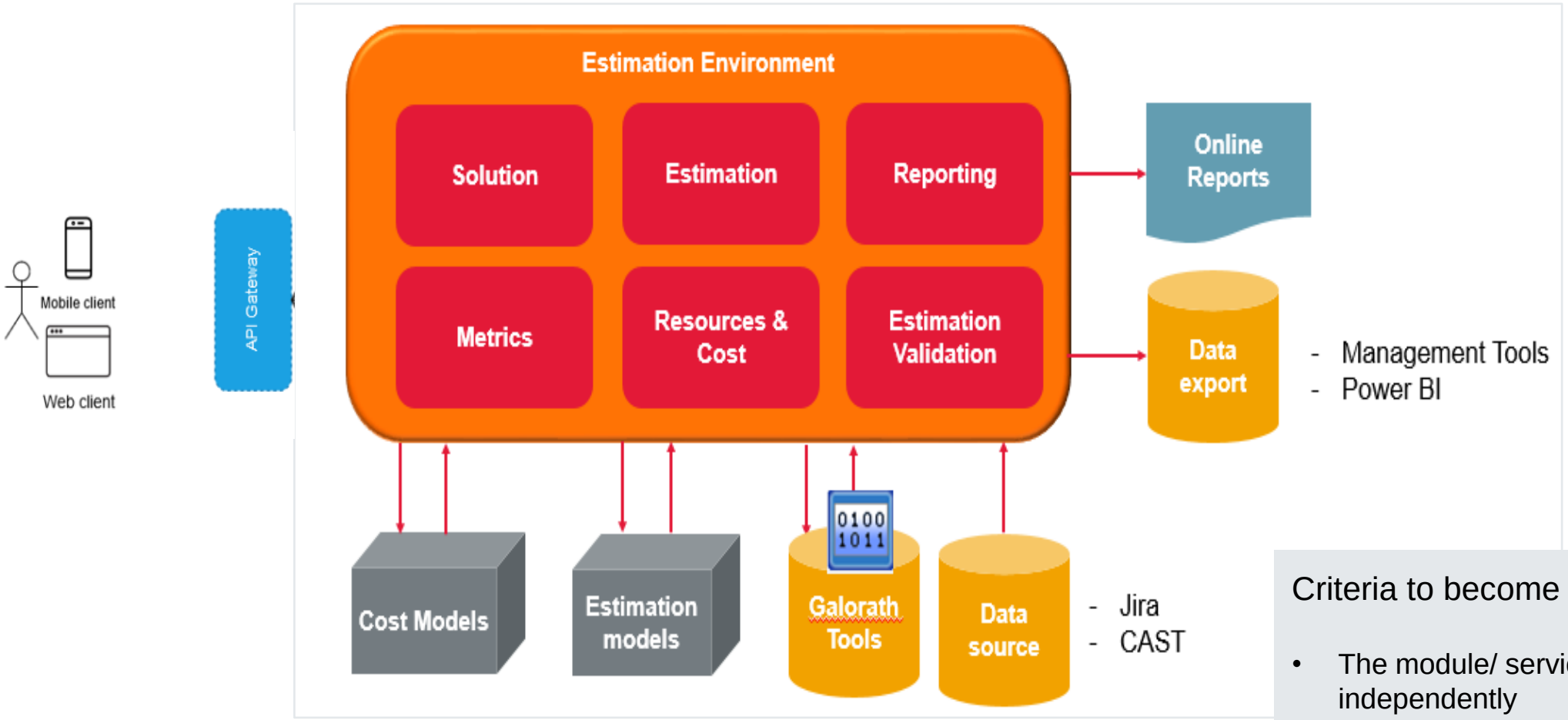
Nesma conference

November 2020

Lets look at the storyline…

The build of a Global Estimation Application

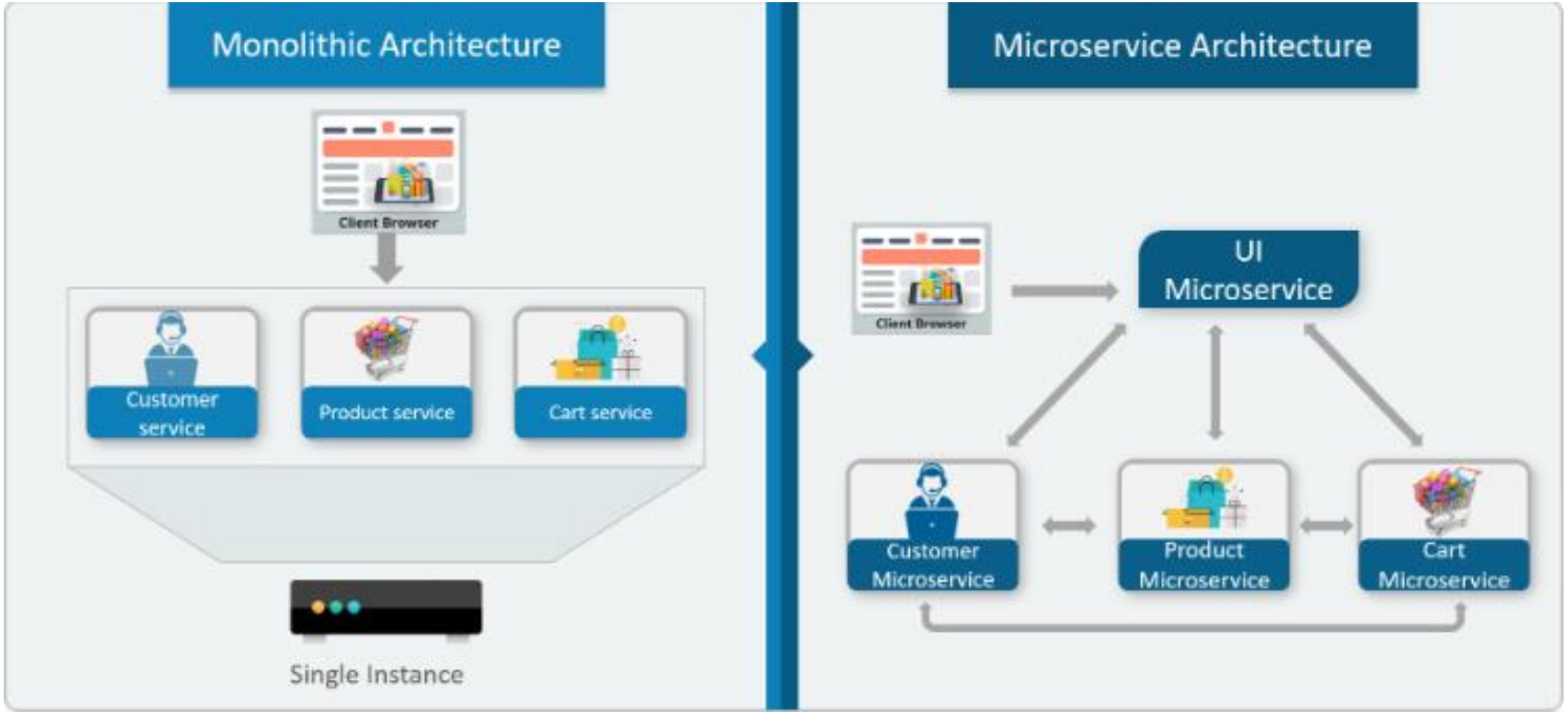
Estimation Application architecture based on Microservices



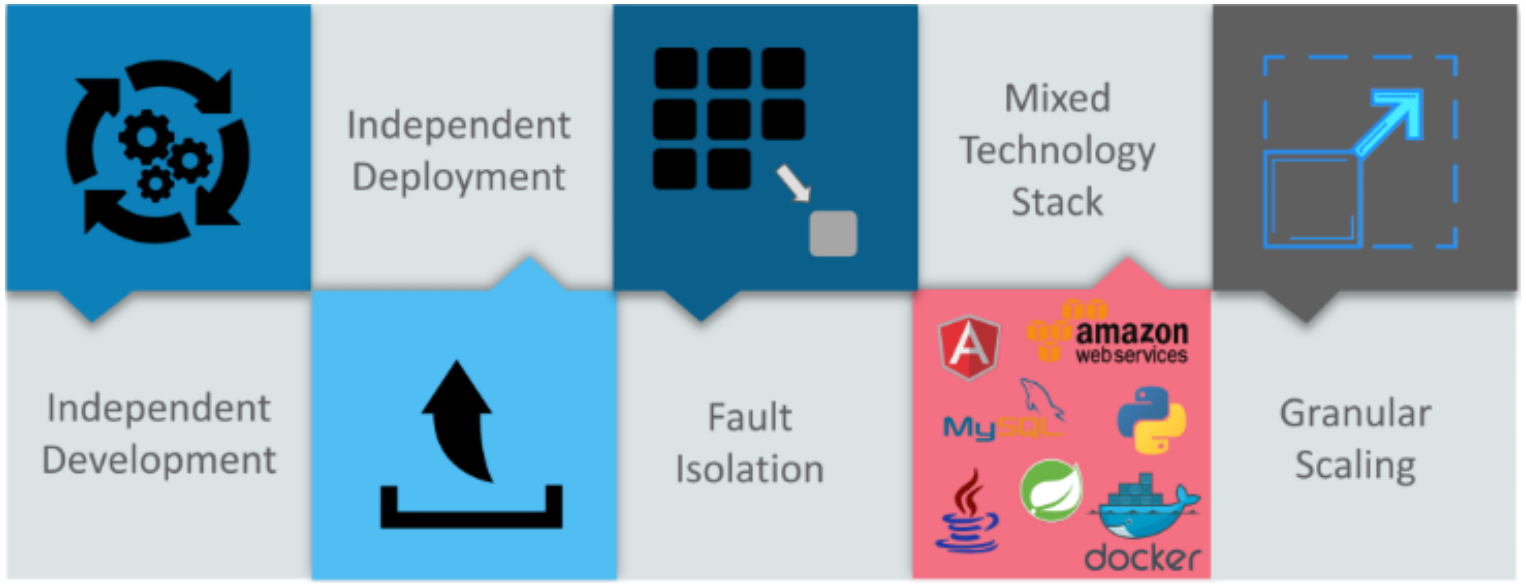
Criteria to become a Microservice:

- The module/ service shall exist independently
- It can be exposed to other applications directly without going through main application

Choice between Monolithic vs Microservices



The advantages of Microservices architecture



- **Independent Development** – All microservices can be easily developed based on their individual functionality
- **Independent Deployment** – Based on their services, they can be individually deployed in any application
- **Fault Isolation** – Even if one service of the application does not work, the system still continues to function
- **Mixed Technology Stack** – Different languages and technologies can be used to build different services of the same application
- **Granular Scaling** – Individual components can scale as per need

Cost Estimation & Verification – The case

- The tool is a new development following an Agile lifecycle
- 18 months of development has been completed; another 6 months to go
- A production ready Minimum Viable Product has been delivered
- Management wants to understand if we are on track with respect Cost & Schedule

<u>The cost estimation process applied:</u>

- Sizing the application with the delivered functionality – **CGI Quick Function Point approach (QFP) has been used**
- Use of actual performance of development team – **Data from timesheet / JIRA and other available sources**
- Verification of estimates with ISBSG data – **ISBSG data set was used in conjunction with Galorath SEER SEM**
- Apply rate card to determine the cost and check this against the budget – **Publish the results**

Function Point sizing

- QFP tool uses Proxies which are technology specific. The output correlates with IFPUG/ NESMA Function Point size with variation of not more that 8-10%
  - Detailed method on QFP and validated results were published at ISMA 13 conference

- List of **Generic Proxies** that can be used in most of the situations

Logical Data Entity
External Logical Data Entity

Data
Default weightage : Low

User Input Interface
Search / View / Query
Reports
Background process
Message
Web Services
Interface

Transaction
Default weightage : Average

Cost Estimation – Initial approach

Sizing Approach:

- It considered the complete application within the boundary

Effort estimates verification:

With the size, estimate was performed with key considerations:

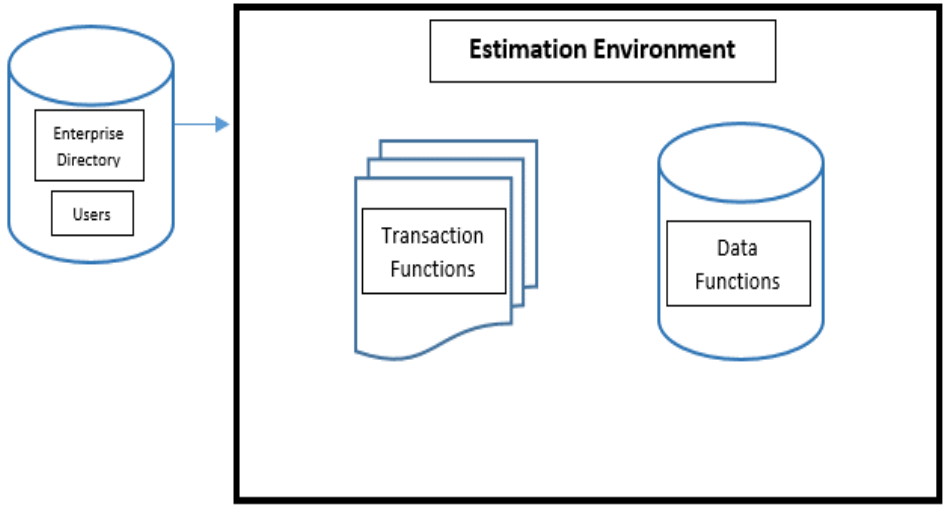
- Front end has a high complexity
- High level of security requirements for data confidentiality

Actuals on Estimate and Schedule

- Collected and verified by multiple means i.e. Timesheet, JIRA etc.

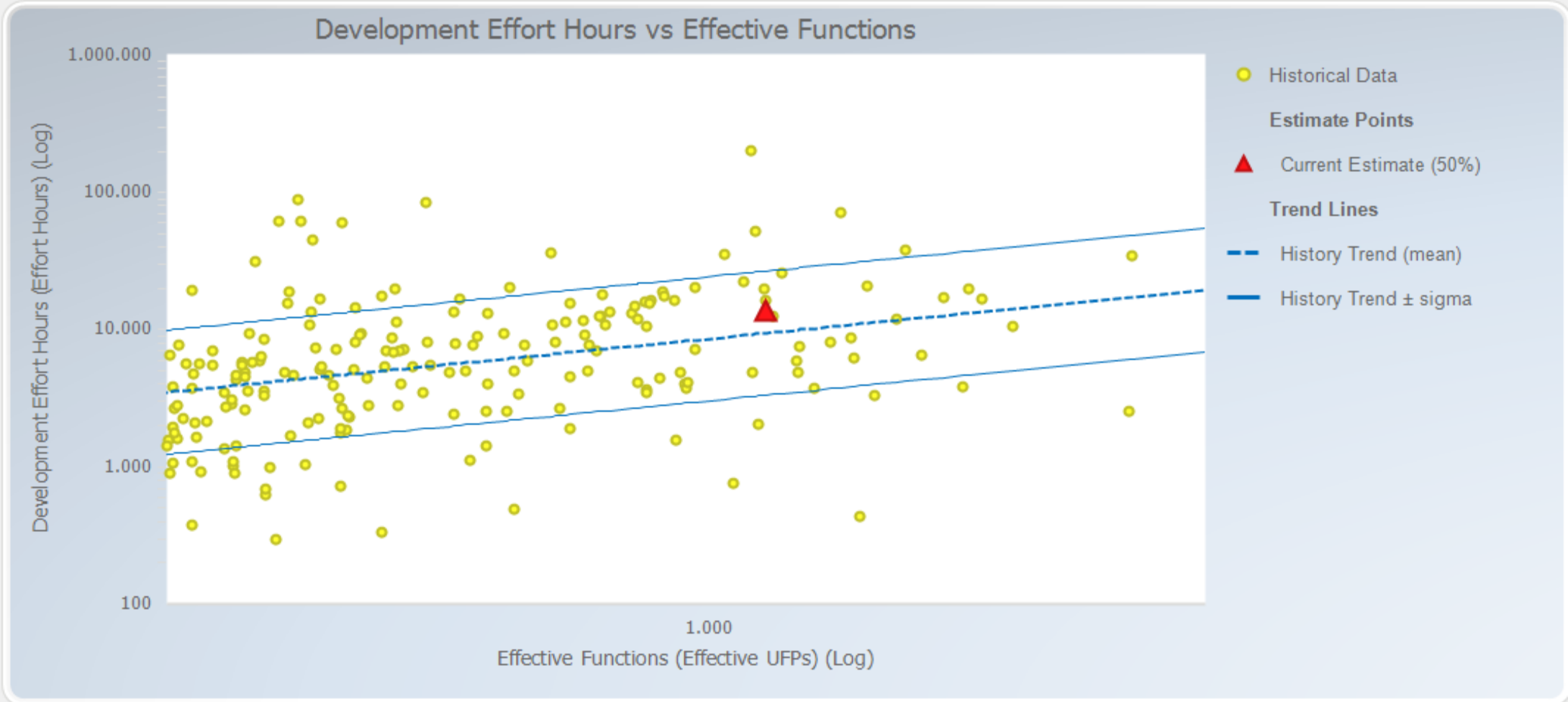
Cost estimates vs actuals comparison

- Difference of **34%** found between the Estimates and Actuals



There was a serious amount of difference between the effort and the actuals that needed further investigation

Verification with ISBSG dataset



- Based on the estimated size we've calculated a trend based estimate
- An explanation was required for the difference with the actual effort that was high

Gap identification

- ✓ Actuals on effort and schedule reported by the team verified by the Product Owners
- ✓ Process of booking the time on project verified by the Project Managers
- ✓ Relooking in the Functional size to see if requirements are missed

* Publications let us to find more about Micro services architecture
  We've discussed the implementation mechanism with the architects

Each Microservice

- is a stand alone application
- has its own database
- interacts with the other applications independently

Cost Estimation – Revised approach for functional sizing

Sizing Approach:

- Each micro service considered independently as an application
- **35%** increase in data size mainly due to rise in External Interface Files
- **20%** increase in Transaction size
  - Increased complexity driven by File Type References (FTRs)
  - Number of transactions due to increased communication between applications
- **21%** increase in functional size for the overall application

Effort estimate verification:

With the size, estimate was performed with key considerations:

- Front end has a high complexity
- High level of security requirements for data confidentiality organization

Actuals on Estimate and Schedule

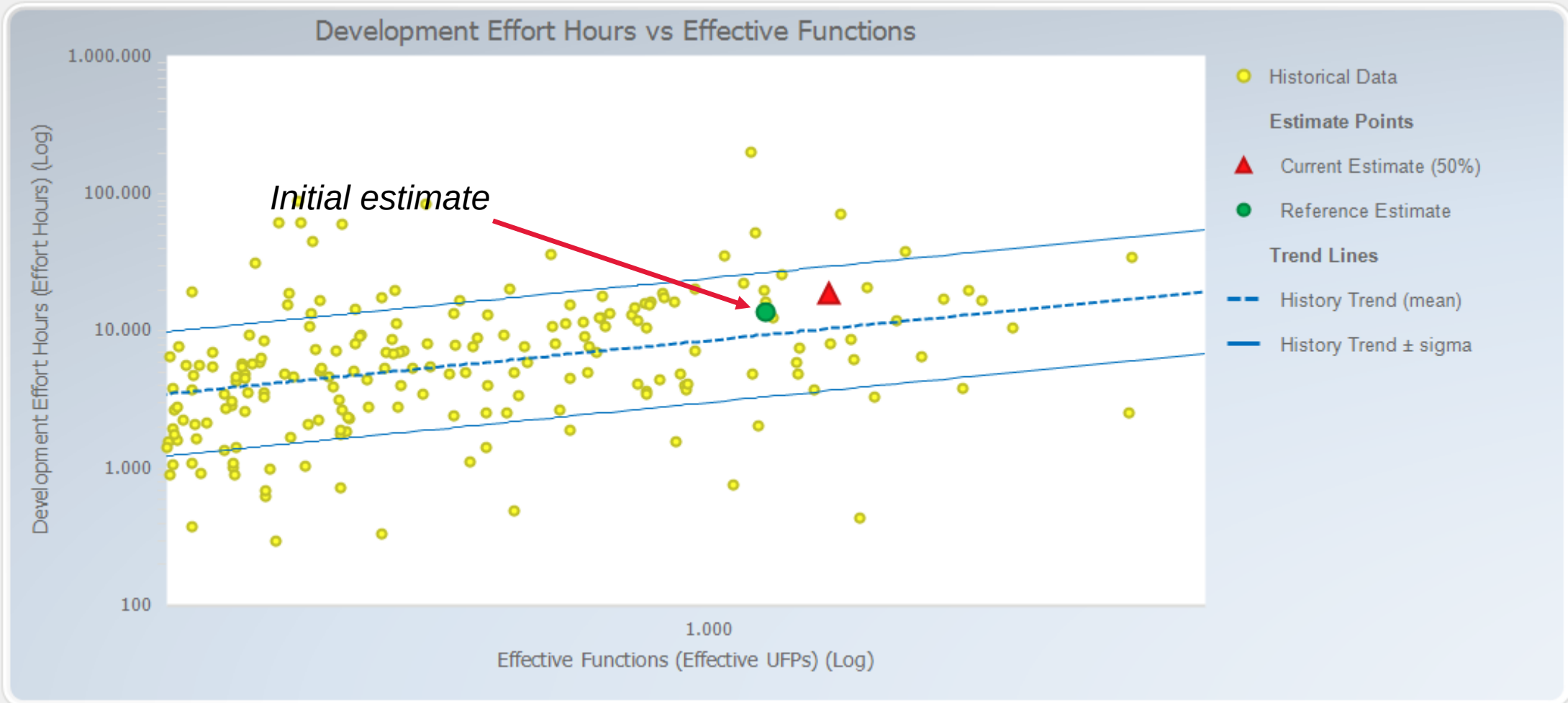
- Collected and verified by multiple means i.e. Timesheet, JIRA etc.

Cost estimates vs actuals comparison

- Difference of **2%** found between the Estimates and Actuals

There is a need to look at the sizing approach for micro services based application to achieve a reliable cost estimates for project

Verification with ISBSG dataset



- The size is higher and with a correction of complexity parameters the effort becomes higher
- The amount of ISBSG data in the defined range is limited but the estimate is still within 1 sigma

Conclusion

- Microservices based application are more complex than a monolithic architecture
- The cost estimation of a microservice based application requires a different sizing approach
- The applied sizing approach impacts mainly the Data functions (less the Transaction functions)
- With the adjusted size and complexity parameters we were able to create an accurate estimation model that also could be applied to forecast the remaining work
- Sizing of microservice based applications require further guidelines to achieve consistency

References

- ISMA 17 Published paper “Transformation from Monoliths: How to Size?” by Saurabh Saxena & Kiran Yeole

- https://www.edureka.co/blog/what-is-microservices/

- https://developer.ibm.com/depmodels/microservices/

Questions?

Thank You

Bhawna Thakur

E-mail: Bhawna.Thakur@cgi.com

Linked-in: www.linkedin.com/in/thakurbhawna