Experiment-as-a-Service in the Pipeline: Empowering CI/CD with xG Acceptance Testing

Sergio Barrachina-Muñoz, Horacio Bleda, Manuel Requena, Selva Vía, Miquel Payaró, Josep Mangues-Bafalluy
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain
{sbarrachina, hbleda, mrequena, svia, miquel.payaro, josep.mangues}@cttc.cat

***Abstract*—Experimentation in 5G and next-generation (xG) mobile networks is crucial for overcoming the limitations of theoretical models and simulations in accurately reflecting real-world complexities. The dynamic nature of advanced wireless communications necessitates practical validation through hands-on testing, a need increasingly addressed by 5G and beyond testbeds. However, challenges remain in ensuring these testbeds are easy to use and open enough to enable users and companies to interact autonomously with the experimental platforms, minimizing human intervention towards a zero-touch approach. This work, presented as a short paper on ongoing research and innovative ideas, introduces a novel approach that leverages Experiment-as-a-Service (ExaS) to enhance Continuous Integration and Continuous Deployment (CI/CD) by integrating xG testbeds into the acceptance testing phase. We discuss the challenges of enabling automated interactions between CI/CD pipelines and xG testbeds. Finally, we present a use case to illustrate the vision of the paper by mapping each phase of the CI/CD pipeline to the example use case, highlighting the benefits of integrating ExaS into the acceptance testing phase.**

***Index Terms*—testbed, experiment-as-a-service, CI/CD, 5G, xG**

I. Introduction

Experimentation in 5G and next-generation (xG) mobile networks is essential due to the limitations of theoretical models and simulations in capturing real-world complexities. Advanced wireless communications are shaped by factors such as environmental conditions, device heterogeneity, and unpredictable interference, necessitating practical validation through hands-on testing. While 5G and beyond testbeds have been developed to address these needs, accessibility and usability remains a challenge. Experiment-as-a-Service (ExaS) is emerging as a solution, providing a framework for efficiently defining, executing, and managing experiments. ExaS aims to streamline workflows, ensuring data integrity, timely results, and increased automation, ultimately accelerating innovation in both academic and industrial environments.

In our vision, with ExaS as foundational concept, xG testbeds can now be regarded as third-party resources integrated into CI/CD pipelines for any software company looking to validate new features against close-to-real xG networks. These CI/CD pipelines encompass a set of practices in software engineering aimed at automating the integration of code changes, testing them, and deploying them into production. Further, this is an opportune time for such a vision, as the telco industry is increasingly adopting CI/CD practices to manage its growing complexity of software delivery. Indeed, the shift from monolithic applications to cloud-native, microservices-based architectures is driving this transition, allowing for greater flexibility and scalability in managing telco solutions [1], with applications requiring frequent updates. This telco's shift to CI/CD and the need for rigorous xG testing create the ideal environment for proposing ExaS as an externalized solution for acceptance testing in CI/CD pipelines, which is particularly relevant considering the dynamic operational conditions that wireless networks introduce.

This work, presented as a short paper on ongoing research and innovative ideas, presents a novel vision to enhance the software development processes of organizations by integrating xG testbeds as integral components of the acceptance testing phase within their CI/CD pipelines. We explore the key ideas and associated challenges of enhancing testbeds to accept automated requests from CI/CD pipelines, thereby enabling organizations to conduct targeted experiments to efficiently validate their software developments against xG networks. To structure our discussion, we employ a specific use case of an app that illustrates each component of the CI/CD pipeline, framing how the acceptance testing phase could effectively interface with any xG testbed.

II. The role of ExaS in xG testing

As the telco landscape evolves beyond 5G, experimental platforms like 5GENESIS, 5G-VINNI, and 5G-EVE (and other similar efforts) provided a critical infrastructure for companies, researchers, and public institutions to validate new technologies and test next-generation applications. 5GENESIS [2] offers a large-scale facility for evaluating 5G performance across diverse environments, supporting verticals such as smart cities and autonomous systems. 5G-VINNI [3] enables end-to-end 5G service validation in real-world scenarios, benefiting sectors like healthcare. 5G-EVE [4] accelerates digital transformation by facilitating the evaluation of new 5G technologies in various industries. Further, 6G-SANDBOX [5] focuses on early 6G experimentation, exploring the merge of digital, physical, and human domains.

While these platforms successfully bridge theoretical advancements with real-world deployment, challenges persist in achieving a fully functional ExaS model. Despite their valuable infrastructure, many platforms lack seamless, automated interaction capabilities, which hinders accessibility and user experience. Overcoming this limitation is essential for enhancing experimentation and driving innovation, especially for small and medium enterprises and academic institutions.

This work was partially funded by the "Ministerio de Asuntos Económicos y Transformación Digital" and the European Union-NextGenerationEU in the frameworks of the"Plan de Recuperación, Transformación y Resiliencia" and of the "Mecanismo de Recuperación y Resiliencia" under references TSI-064100-2022-16/-2023-26 (Plaza6G/Plaza6G+), Spanish MCIN I-CERCA CERCAGINYS, and Generalitat de Catalunya 2021 SGR 00770.

ISBN: 978-3-903176-71-3

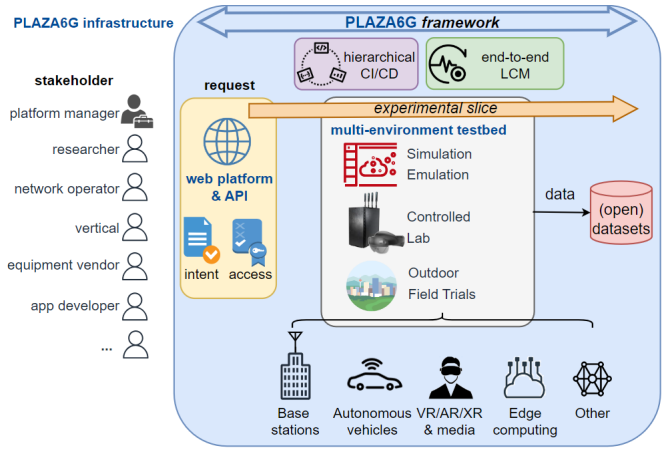

Fig. 1: Overview diagram of the Plaza6G vision.

The Plaza6G [6] initiative recently started with the aim to establish a framework that fundamentally supports ExaS while enabling seamless interaction with a wide range of users, including traditional software organizations (or verticals) keen on assessing their services' performance over xG networks. This innovative platform seeks to provide an outsourced acceptance testing phase within near-real networks, enhancing the software validation process. Plaza6G envisions integrating several critical components based on the Experiment as Code (ExaC) model [7]. As shown in Fig. 1, Plaza6G will offer stakeholders three types of experimental environments - simulation, indoor laboratory, and outdoor testing - to accommodate various stages of technology development and evaluation interests. These components will collectively leverage the ExaC model to create a pioneering ExaS platform [8], [9], enabling users to manage experiments with zero-touch automation, eliminating the need for human intervention.

III. CI/CD for xG Network Acceptance

CI/CD is essential in modern software engineering since it enables organizations to automate and streamline their development processes. Indeed, a recent study found out that, from a sample of 600,000 software repositories (excluding small projects), a third use CI/CD today [10].

A. What is CI/CD?

CI involves the frequent integration of code changes into a shared repository, utilizing automated testing to identify issues early in the development cycle. This encompasses various testing types, including unit testing to validate individual components, integration testing to assess interactions between integrated units, and acceptance testing to ensure the software meets predefined user requirements. CD, conversely, automatically releases software updates into production upon successful testing completion, minimizing manual intervention and accelerating the delivery of new features. In the context of xG networks, Fig. 2 illustrates the typical phases of a CI/CD pipeline, incorporating our proposed interface to the acceptance testing phase to interact with the xG testbed. The figure also highlights the ownership and location of each CI/CD phase, as explained later in §IV.

B. Why integrating xG ExaS into the CI/CD pipeline?

Incorporating an external service for network acceptance testing during the CI/CD phase offers substantial benefits for software organizations, such as mobile application developers, particularly when evaluating applications in next-generation 5G, 6G, or Wi-Fi 8 networks. Many companies lack direct access to advanced wireless network infrastructures, which limits their ability to assess application performance under highly dynamic real-world conditions. As applications increasingly demand ultra-low end-to-end latencies, high bandwidths, and massive device connectivity, evaluating performance in mobile network environments becomes essential for comprehensive assessments. Therefore, software companies must ascertain whether their new features can meet performance requirements in xG networks under specific conditions.

Offering ExaS on xG networks, similar to how cloud platforms provide computing resources, allows companies to leverage external infrastructure for testing applications. This approach enhances the acceptance testing process by evaluating new network features, going beyond traditional unit and functional tests. E.g., conducting acceptance tests on actual smartphones or headsets within live 5G networks provides a more accurate assessment of application performance. Testing in live wireless environments ensures reliable application operation across diverse network scenarios, thereby improving user satisfaction and minimizing post-deployment issues. Additionally, integrating xG network acceptance testing into CI/CD pipelines accelerates development cycles, facilitating faster and more reliable releases of network-dependent features.

C. Challenges

The challenges for integrating ExaS testbeds in the acceptance testing phase of a CI/CD pipeline can be classified in (i) experimental substrate, (ii) usability, and (iii) business context.

The first group pertains to the management of the experimental infrastructure, including availability of sufficient network and computing resources in the testbed. To support various testing scenarios, it must be equipped with capabilities to address different network segments, including next-generation Radio Access Networks (RAN), core networks, and transport segments that can handle high-throughput data transfers. Computing resources are also crucial, potentially incorporating Multi-Access Edge Computing (MEC) services to enhance edge capabilities. A diverse range of end devices, such as smartphones, headsets, and autonomous vehicles, should be integrated to emulate real-world user experiences. Additionally, the testbed must provide controlled environments, such as labs for precise testing of specific variables, alongside outdoor environments that reflect actual operating conditions. Furthermore, isolation of contexts between experimenters throughout the experiment lifecycle must be guaranteed (traffic flows, configuration, KPIs, traces, etc.)

The second group is related to offering an automated easy-to-consume interface for developers to be able to run experiments as if they were using any other conventional library. This entails achieving true autonomy within the testing

ISBN: 978-3-903176-71-3

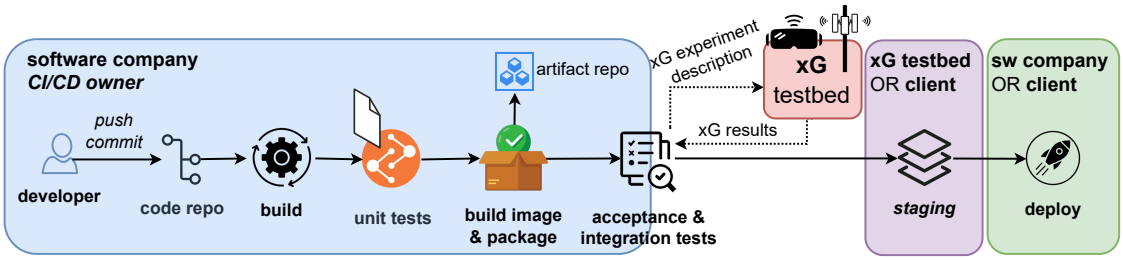

Fig. 2: CI/CD pipeline with an integrated interface for xG testbed acceptance testing, highlighting the ownership of each phase.

process, commonly referred to as zero-touch capabilities. While many existing ExaS implementations fall short of this level of automation, it is crucial for ensuring seamless operation. Efforts such as the Plaza6G framework presented in §II are actively working towards delivering autonomous testing environments that minimize manual intervention. However, until these solutions are fully developed and deployed, attaining a completely autonomous testing experience remains a significant challenge. The third challenge involves the clarity and robustness of the exposed RESTful APIs [11] within the testbed. To ensure seamless integration with CI/CD pipelines, these APIs must be well-defined and user-friendly, facilitating smooth interactions between developers and the testing infrastructure. The trade-off between API simplicity towards ease-of-use and genericity towards experiment definition flexibility must be explored. Comprehensive documentation is also crucial, since well-defined RESTful APIs are expected to provide standardized communication methods, simplifying the integration of applications with testbed functionalities.

Finally, in a broader business context, establishing trust among users is paramount; stakeholders must be confident in the legitimacy, accuracy, and reproducibility of tests conducted on ExaS testbeds to foster greater adoption of these platforms. This trust can be generated through several means: implementing transparent testing methodologies that allow users to understand the processes involved, conducting independent audits to validate the integrity of the testbed operations, and providing comprehensive documentation of past test results, including any anomalies and how they were addressed. Additionally, engaging in collaborative partnerships with industry leaders to endorse the ExaS framework and the creation of open standardized APIs can further enhance credibility.

IV. A Use Case Example

A software company has developed an edge-based Large Language Model (LLM) app for industrial sectors, such as manufacturing. In automotive factories, workers can ask questions to the LLM and receive real-time, context-specific responses, enhancing operational efficiency. While local unit tests could verify software functionality, they would fail to evaluate performance under real-world wireless network conditions. Thus, integrating ExaS testbeds into the CI/CD pipeline allows the company to automate network acceptance tests for new features, focusing on the app's behavior across different scenarios involving xG networks.

A. An LLM Application for Automotive Industry

The app, developed using Android Studio, communicates with a Python-based LLM server located at the network edge within the manufacturer's premises via a RESTful API. This enables workers to receive real-time, AI-generated guidance on manufacturing processes, with responses delivered in text, audio, or video formats. Such interaction demands a high-performance experimental substrate, including networking and computing resources. As for the network, very low latency and, in the case of video, substantial throughput, are required. When a worker submits a query via the app, the request is sent to the Python server over a designated port (e.g., 65000), where it is processed using a pre-trained LLM model.[1] The LLM generates a response, providing real-time support for troubleshooting, operational guidance, or decision-making.

Remarkably, the server's CPU (or GPU) utilization can become substantial due to the LLM server processes, making it a critical factor to incorporate into the network acceptance testing phase. With xG ExaS integrated into the CI/CD pipeline, not only can the xG network conditions be tested, but also the server's performance across various computing environments, including physical, virtualized, or containerized systems. These computing resources can be provisioned by the xG ExaS testbed itself, as exemplified by Plaza6G, which exposes computing capabilities through its API that can be requested through experiment descriptors, allowing comprehensive evaluation of both network efficiency and server load under realistic conditions.

B. CI/CD steps of the use case

1) From commit to package: The CI/CD pipeline is triggered automatically when a developer commits a change to the code of the app.[2] For instance, the new feature in this case is the addition of a "speak" functionality, allowing users to not only type, but also voice their questions to the server (see Fig. 3). Following the commit, the code is built into an executable format, involving steps such as downloading dependencies, installing necessary tools, and performing the compilation process. Then, automated unit tests are executed to validate the functionality of individual components, verifying that the "speak" feature operates correctly alongside the existing "write" functionality. Unit tests are performed locally

[1]We utilize the distilgpt2 model, though other models could be employed.
[2]Notice that the same or a different CI/CD pipeline will also be triggered for any changes made to the LLM server code.

ISBN: 978-3-903176-71-3

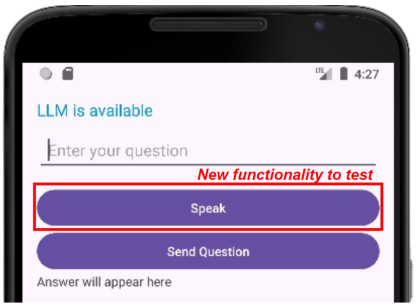

Fig. 3: App user interface with new speak feature highlighted.

and can include, among others, validating user input length to ensure it does not exceed maximum limits, verifying correct token count calculation for LLM constraints, checking proper inclusion of the API key in request headers, ensuring user input is accurately serialized to JSON format, or testing the rate-limiting mechanism to prevent excessive requests to the LLM server within a specified timeframe. Once the unit tests pass, a container image of the app is normally created and pushed to the company's artifact repo, encapsulating the code and its dependencies for deployment in various environments.

2) Acceptance and integration test: Integration tests focus on verifying the functionality of combined modules within a single code base, and are typically written by the engineers responsible for the code. In contrast, acceptance tests evaluate the entire product (including the network) and are generally developed and executed by a separate team.

In our proposed vision, acceptance testing can be classified into two categories: local acceptance tests and externalized acceptance tests. Local acceptance tests are conducted by the CI/CD owner. In the use case under discussion, these tests could include validating the end-to-end message flow from user input to the response generated by the LLM server, ensuring graceful handling of large inputs, and verifying that the app presents user-friendly error messages during server downtimes. Additional tests assess the app's ability to detect input languages and respond appropriately, manage user sessions effectively, handle multiple simultaneous requests without performance degradation, and ensure data privacy through encryption. Furthermore, stressing the loads tests the app's capability to maintain functionality during high demand.

Conversely, externalized acceptance tests are conducted in an xG testbed, which is specifically prepared to receive and process experiment requests due to its ExaS nature. This testbed supports efficient communication with the CI/CD pipeline through a dedicated set of APIs. In this use case, the xG testbed enables comprehensive evaluations of both communication (network) and server latency. Network latency tests examine various configurations across 4G, 5G, and xG technologies, ensuring that the app can reliably send requests and receive responses from the LLM server while providing users with timely feedback regarding potential delays. Additionally, server latency tests explore scenarios involving CPU and GPU resource utilization, emphasizing the effects of resource allocation on response times. If the acceptance test passes, the pipeline proceeds to the staging phase. However, if the test fails, the xG testbed provides detailed feedback on why it failed, identifying which specific Key Performance Indicators (KPIs) did not meet the required thresholds.

3) Staging and Deployment: Staging is the phase where the app is deployed to a pre-production environment for testing and validation to ensure its functionality and performance meet the required standards. In our LLM app use case, staging can occur either in the xG testbed or at the manufacturing premises during off-hours, such as nighttime when operations are paused, allowing for comprehensive evaluation without disrupting ongoing activities. Deployment, on the other hand, refers to the final phase where the validated app is released to the production environment, making it accessible to users in real-time. For the LLM app, this means deploying it within the manufacturing premises to facilitate immediate access for workers seeking guidance on manufacturing processes.

V. Conclusions

This paper presents a vision for the integration of ExaS into CI/CD frameworks for 5G and xG networks, emphasizing the need for practical experimentation to complement theoretical models. We identify the challenges that must be addressed to enable seamless, zero-touch interactions between CI/CD pipelines and xG testbeds, highlighting the necessity for user-friendly platforms with robust APIs for autonomous testing. By incorporating ExaS into the acceptance testing phase, we aim to enhance software validation efficiency and reliability, as suggested by our use case that demonstrates the potential for effectively mapping CI/CD phases to optimize application performance in real-world conditions. We envision a framework where any software company can seamlessly test advanced networking and computing capabilities through a fully automated, third-party platform in a zero-touch process.

References

[1] "CI/CD: Continuous software for continuous change," Ericsson, Tech. Rep., 2022. [Online]. Available: https://www.ericsson.com/en/core-network/guide/forms/guide-ci-cd

[2] G. Xylouris *et al.*, "Experimentation and 5G KPI measurements in the 5GENESIS platforms," in *Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases*, 2021, pp. 1–7.

[3] A. J. Gonzalez, M. Xie, P. H. Lehne, and P. Grønsund, "Achieving high throughput and low latency with 5G: A real implementation experience," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 84–90, 2021.

[4] M. Gupta *et al.*, "The 5G EVE end-to-end 5G facility for extensive trials," in *2019 ICC workshops*. IEEE, 2019, pp. 1–5.

[5] P. Merino-Gomez, B. Garcia, C. Andreo, D. Artuñedo, and J. Macias, "On-demand Trial Networks over 6G-SANDBOX infrastructure," in *2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2024, pp. 1021–1026.

[6] "PLAZA6G webiste," https://www.plaza6g.eu/.

[7] L. Aguilar, M. Gath-Morad, J. Grübel, J. Ermatinger, H. Zhao, S. Wehrli, R. W. Sumner, C. Zhang, D. Helbing, and C. Hölscher, "Experiments as Code and its application to VR studies in human-building interaction," *Scientific Reports*, vol. 14, no. 1, p. 9883, 2024.

[8] T. W. Edgar and T. R. Rice, "Experiment as a service," in *2017 IEEE Int. Symp. on Tech. for Homeland Security (HST)*. IEEE, 2017.

[9] M. Boniface *et al.*, "BonFIRE: A Multi-Cloud Experimentation-as-a-Service Ecosystem," in *Building the Future Internet through FIRE*. River Publishers, 2022, pp. 243–266.

[10] H. Da Gião, A. Flores, R. Pereira, and J. Cunha, "Chronicles of CI/CD: A Deep Dive into its Usage Over Time," *arXiv preprint*, 2024.

[11] A. Golmohammadi, M. Zhang, and A. Arcuri, "Testing RESTful APIs: A survey," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 1, pp. 1–41, 2023.

ISBN: 978-3-903176-71-3