



Implications of the Magma Architecture

Interoperability, scale
and resilience

Executive Summary

Magma is one of several open source implementations of a mobile core with support for LTE and 5G radio access. Its architecture differs from most other implementations in important ways. Standard 3GPP protocols are supported at key interfaces, notably to the RAN (radio access network) and the Federation interface to other mobile networks. However, unlike typical mobile core implementations, Magma does not support 3GPP interfaces between the components that make up the mobile core. This architecture has implications for interoperability and testing, while also providing a greater ability to scale (in terms of performance and number of users) than other approaches. Magma also achieves high reliability in certain challenging environments.

Interoperability. Interoperability with standard RAN (LTE or 5G) equipment is supported, as is federation with existing mobile core networks. But because Magma does not implement the 3GPP interfaces that are *internal* to a mobile packet core, it is not possible to arbitrarily mix and match components within the core. A traditional 3GPP implementation would permit (say) an MME from one vendor to interoperate with the S-gateway of another vendor, but it is not possible to connect parts of a mobile core from another vendor (or another open source project) with parts of Magma. Whether this matters depends on the use case.

Testing. Some industry-standard test equipment depends on the existence of internal 3GPP interfaces. Since Magma lacks these internal interfaces, testing with third party tools must be done either using the external interfaces to the RAN or the Federation interfaces. Further testing is achieved using custom tools developed specifically for Magma.

Reliability and Resilience. The Magma architecture terminates 3GPP protocols and runs a local control plane in access gateways, which are located close to the RAN. This eliminates the need to run 3GPP protocols across backhaul links, which improves reliability and resilience to a variety of failure scenarios. It also enables the use of less reliable backhaul links (e.g. over satellite) without a negative impact on overall system reliability. Supporting wireless access in situations of poor quality backhaul has emerged as a key use case for Magma.

Scale-out. The Magma architecture is inspired by software-defined networking, with a central point of control (known as the orchestrator) and a distributed, scale-out implementation of the data plane. This enables a Magma core to scale out gracefully with the growth of radio access points and users by adding access gateways, growing in performance while maintaining ease of operations through a single, central control point. This is a significant point of differentiation from other open source mobile core implementations. Magma has been deployed at a scale of thousands of access gateways controlled by a single orchestrator.

Contents

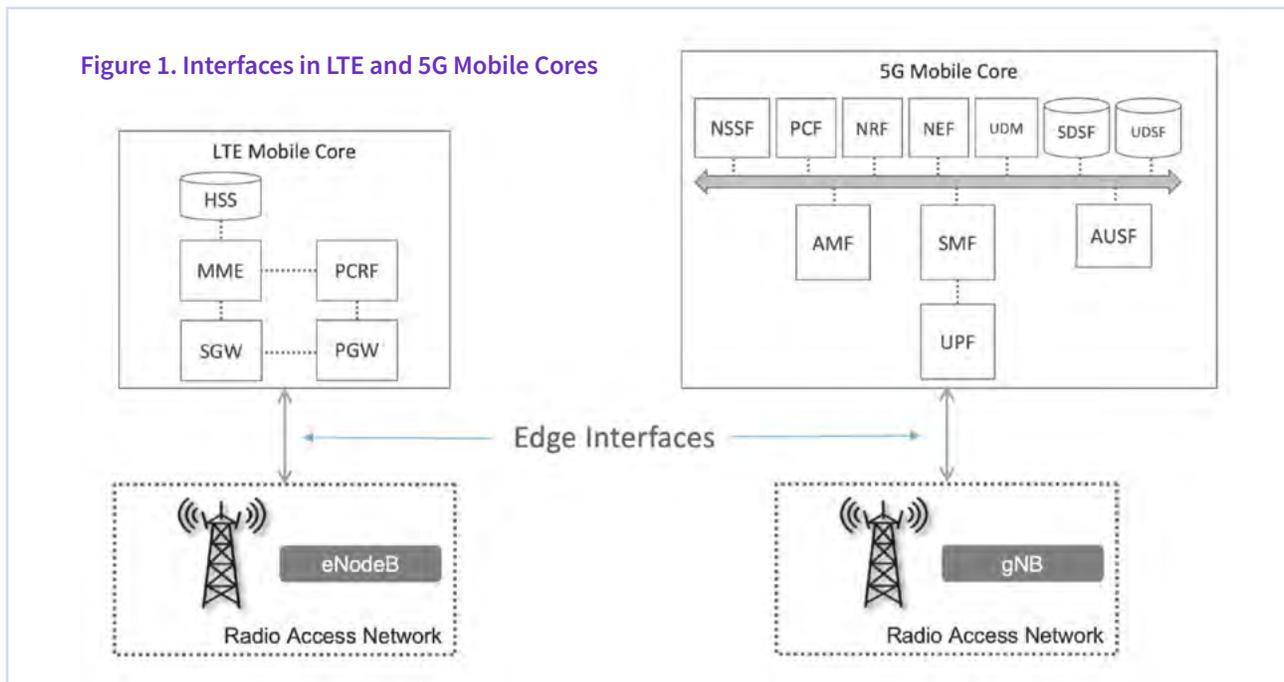
Executive Summary	2
Introduction	4
Edge Termination of Protocols	4
Standards Compliance	6
Testing Implications	7
Reliability of Backhaul	7
Resilience	7
Central Control with Scale-out Implementation	8
Performance	9
Mobility across Gateways	9
Conclusion	10

Introduction

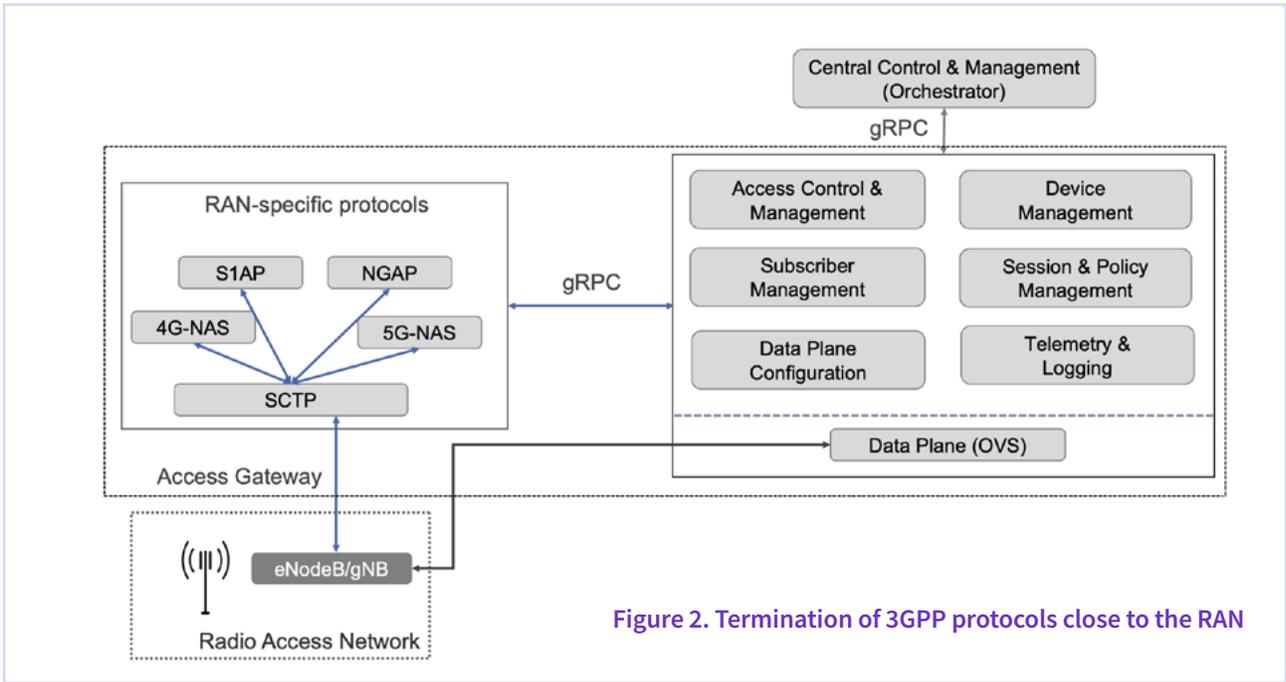
There are numerous open source implementations of mobile packet cores, e.g. [Open5GS](#), [Free5GC](#) and [OMEC](#), and Magma is yet another such implementation. However, Magma differs from these others in one key architectural respect: Magma terminates 3GPP protocols logically close to the “edge”, where edge in this context means either the radio interface or the federation interface to another mobile network. The single architectural decision was not made casually, and it has broad implications, which we explore in this post. A more detailed discussion of the Magma architecture and experiences is available [here](#).

Edge Termination of Protocols

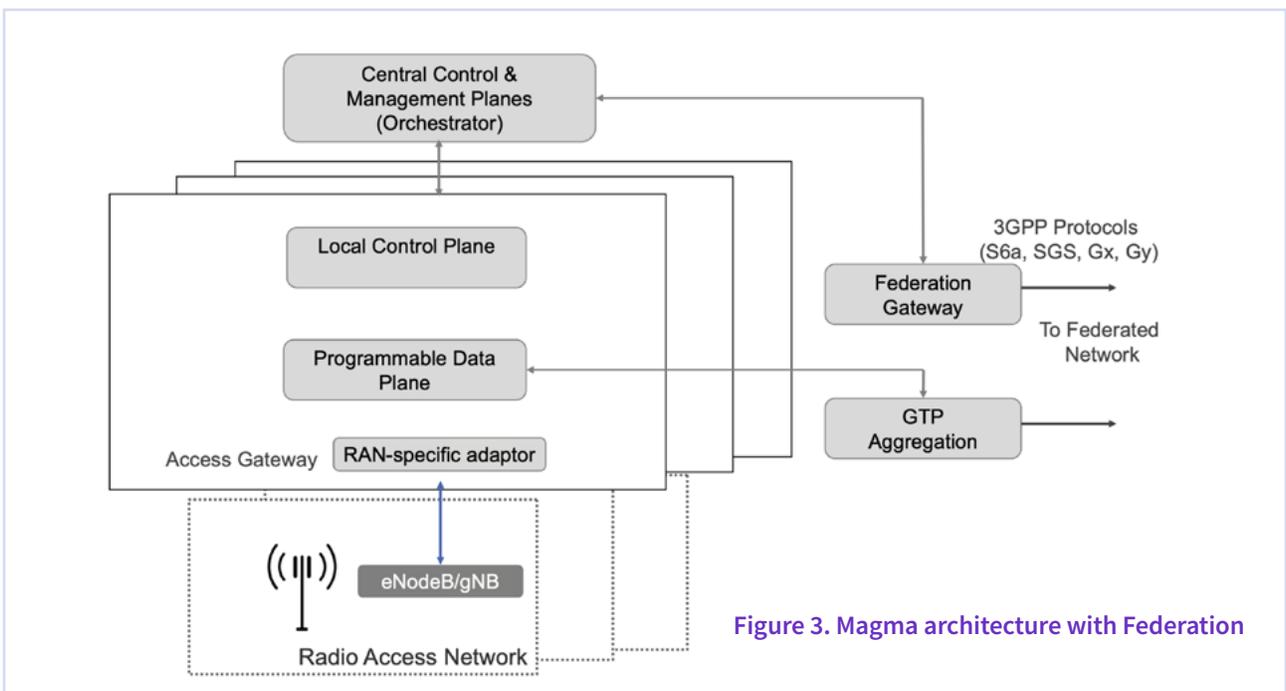
Let’s start by explaining what we mean by “edge termination”. The 3GPP standards define interfaces among all the components of the RAN (Radio Access Network) and the mobile core, whether it is the EPC (evolved packet core), in the case of LTE, or the 5GC (5G core).



Every line between a pair of boxes in Figure 1 is some sort of interface defined as part of the [3GPP Specifications](#). The ones of most interest to Magma are those labeled “Edge Interfaces”, since these are the ones that operate between the RAN and the mobile core. Magma implements the protocols that run over these interfaces in compliance with the 3GPP standards. Internally, Magma terminates 3GPP protocols as logically close to the RAN as possible as illustrated in Figure 2.



To take a concrete example, the S1AP interface as defined by 3GPP runs between an eNodeB and the MME (Mobility Management Entity). S1AP messages are delivered over SCTP (a reliable transport protocol). In Magma, however, the S1AP interface terminates on a module that sits next to the eNodeB. The S1AP messages are translated into gRPC messages, and all further communication among Magma components is via gRPC. This decision to terminate all the protocols specific to the RAN as close to the RAN as possible has wide-ranging implications as discussed below.



Magma also terminates 3GPP protocols on its other edge, the Federation Gateway, as shown in Figure 3. This is the point at which a Magma deployment can be interconnected with an existing mobile operator's core, and the various interfaces implemented here enable the billing and subscriber information in the existing core to be shared with the Magma deployment. Again, the Federation gateway terminates those 3GPP protocols at the first opportunity—inside the Federation Gateway—and then passes all necessary information to the rest of Magma using gRPC.

In summary, there are two “edges” to Magma where 3GPP protocols are implemented: the interface to the RAN, and the interface to federated networks. Thus Magma maintains 3GPP compliance at these points and interoperates with standard equipment running on the other side of such interfaces. But when we look inside Magma, we see all internal communications taking place over gRPC, and not standard 3GPP protocols.

Standards Compliance

The decision to implement standard 3GPP protocols only at the edge means that Magma can interoperate with other implementations only at the edges. Thus, it is possible to connect a Magma mobile core to any standards-compliant eNodeB or gNB and expect it to work. Similarly, a Magma core can be federated with an existing MNO's LTE network. Support for a specific feature of LTE or 5G might sometimes be a matter of the development roadmap, but architecturally, interoperability with any standard system is expected at these interfaces.

Conversely, since Magma does not implement all the 3GPP interfaces that are internal to a mobile packet core, it is not possible to arbitrarily mix and match components within the core. Whereas a traditional 3GPP implementation would permit (say) an MME from one vendor to interoperate with the S-gateway of another vendor, it is not possible to connect parts of a mobile core from another vendor (or another open source project) with parts of Magma aside from via the interfaces described above. Whether this matters depends on the use case.

The Telecom Infra Project's [Open Core Network](#) (OCN) Project Group is defining a set of technical requirements for mobile core networks that are based on specific use cases such as fixed wireless access and private 5G. Establishing compliance with these requirements is part of the ongoing Magma effort.

The decision not to implement all of the internal interfaces of 3GPP in Magma was made consciously to improve the performance and reliability of Magma in a variety of scenarios as described below. It also allows Magma to support a wide range of access technologies (LTE, 5G, and WiFi) from a common, converged software core, lowering implementation complexity. This architectural choice also has implications for both testing and mobility support as discussed below.

Testing Implications

There is an abundant supply of test equipment for performance, scale and compliance testing of 3GPP protocols. For those interfaces where Magma implements 3GPP protocols (the interface to the RAN and the Federation interface) these test tools can be used effectively. Conversely, all the internal interfaces that are implemented using gRPC need to be tested using methods developed specifically for Magma. At a minimum, this increases the effort needed to perform testing of Magma scale and features. Depending on the level of investment in testing, it may also lead to less test coverage of the internal interfaces.

Reliability of Backhaul

Termination of 3GPP at the edge impacts the reliability of the mobile core in a number of ways. Running 3GPP protocols over long backhaul links has proven problematic when the reliability of the backhaul links is less than perfect, e.g., when satellite is used for backhaul. This is because the 3GPP protocols are in some cases quite sensitive to loss and latency. Loss or latency can cause connections to be dropped, which in turn forces UEs to repeat the process of attaching to the core. In practice, not all UEs handle this elegantly, sometimes ending up in a “stuck” state.

Magma addresses the challenge of unreliable backhaul in two ways. First, Magma frequently avoids sending messages over the backhaul entirely by running more functionality in the access gateway. This results from the highly distributed architecture of Magma, where functions that would be centralized in a standard 3GPP implementation are distributed out to the edge (access gateways) in Magma. Thus, for example, the operations required to authenticate and attach a UE to the core can typically be completed using information cached locally in the AGW, without any traffic crossing the backhaul. We return to this topic below. Secondly, when Magma does need to pass information over a backhaul link (e.g. to obtain configuration state from the orchestrator), it does so using gRPC, which is designed to operate reliably in the face of unreliable or low latency links.

Resilience

A significant design choice in Magma is the use of a “desired state” model for runtime and configuration state. By this we mean that to communicate a required state change (e.g., the addition of a new session in the data plane), the desired end state is set via an API. This contrasts with a “CRUD (Create, Read, Update, Delete)” interface, which is common in 3GPP specifications. The desired state model achieves higher reliability in the face of failures, including the failure of components and of communication paths. This is best explained via an example.

Consider the case of establishing data-plane state for a set of active sessions for the UEs served by a single AGW. Suppose there are two active sessions, X and Y. Then a third UE becomes active and a session Z needs to be established. In the CRUD model, the control plane would instruct the data plane “add session Z”. The desired state model, by contrast, communicates the entire new state: “the set of sessions is now X, Y, Z”.

The CRUD model is brittle in the face of failures. If a message is lost, or a component is unavailable for some time and unable to receive updates, the receiver of updates falls out of sync with the sender. So it is possible to end up in a state where the control plane believes that sessions X, Y and Z have been established, while the data plane only has state for X and Y. By sending the entire desired state, we ensure that the receiver comes back into sync with the sender once it is able to receive messages again. This leads to higher resilience to failure of components, both hardware and software. For example, software components in Magma gateways can be restarted (after either crash or upgrade) without a requirement to restart UE sessions, in contrast to traditional core implementations.

Two points are worth noting in regard to the desired state model. First, there are many internal interfaces between modules in Magma, and the desired state model has been fully implemented for just a subset of them. Significantly, the interfaces that run across backhaul links, i.e., interfaces between AGWs and the Orchestrator, adhere to the desired state model. Thus, for example, the “source of truth” for any configuration state is in the Orchestrator, and that state is pushed out to the AGW as needed. There remain some internal interfaces between subsystems inside the AGW that do communicate using state changes.

Secondly, there are scaling implications to sending the complete desired state between two modules when only a small amount of state has changed. As Magma deployments increase in size and subscriber counts increase, it becomes necessary to be more efficient in sending state updates between modules. This has become apparent in sending the subscriber information from the orchestrator to the AGWs. There are many options for improving the efficiency of such updates while retaining the desired state approach, and these efforts (described [here](#)) are ongoing.

Central Control with Scale-out Implementation

An interesting consequence of the design approach taken by Magma is that it allows centralized control to be combined with a highly distributed implementation. This approach was pioneered by the Software-Defined Networking movement, and Magma adopts an SDN-style architecture. For example, while the task of attaching a UE to the core can be completed by local operations at the AGW located next to the base station, as noted above, the operator is not faced with the task of independently managing hundreds of AGWs. Instead, there is a centralized orchestrator which provides both a UI and an API by which an operator (or another piece of software) can interact with the network as a whole. Communication between AGWs and the central orchestrator takes place over gRPC, and there is no comparable 3GPP protocol that could do this job because the standard core architecture of 3GPP does not follow an SDN approach. 3GPP does specify Control and User Plane separation, but there is no concept of a centralized controller with distributed access gateways in 3GPP. This point is discussed further in [an earlier blog](#).

To understand the impact of this decision, consider an operator trying to provide cellular services in remote areas where backhaul is of low quality. It might be tempting to install a traditional EPC next to the radio tower to avoid running the 3GPP protocols across the backhaul link. Any attempt to scale up this solution leaves the operator managing a large number of remote EPCs with no straightforward way to manage the network as a whole. It is precisely this use case—remote access in areas with limited backhaul options—that is driving much of the interest in Magma today.

As a concrete example, if a subscriber is added to Magma via the orchestrator, that subscriber information is propagated to all AGWs. By contrast, in the scenario where multiple traditional EPCs are deployed, each one would need to be configured individually to add the new subscriber.

Performance

The SDN-style architecture of Magma provides a natural advantage in performance. Each time an AGW is added, new data plane and control plane processing capabilities are added to the network. Thus Magma's performance naturally scales up as base stations are added (with each AGW typically supporting a handful of base stations).

As of the most recent tests, we observed a single Magma AGW supporting about 400Mbps in the data plane and 600 concurrent subscribers for LTE, with 400Mbps data plane and 200 concurrent subscribers for 5G. While these represent point-in-time performance numbers, both data rates and subscriber counts smoothly scale up as more AGWs are added to the network. By comparison, our testing of Open5GS shows a single node EPC can support about 95Mbps and 200 concurrent subscribers (for LTE). The difference between these raw numbers is less important than the fact that Magma's data plane and control plane performance scale with the number of AGWs deployed. Test results compiled from a live deployment for a [recently submitted conference paper](#) show Magma's ability to scale up to at least 5,000 AGWs under the control of a single orchestrator.

Mobility across Gateways

In Magma's current implementation, all the runtime state associated with a UE is localized to the AGW to which that UE is attached. While beneficial for scalability, this approach does introduce trade-offs. In particular, supporting seamless mobility across AGWs, to allow a UE to maintain a connection to the core when roaming to another AGW, would require communicating some control-plane state from one AGW to another during hand-off. This is not implemented today; a UE can only roam seamlessly among the set of base stations served by a single AGW (typically fewer than 10 base stations). While many use cases can be supported without this feature, it is a limitation. Members of the Magma community have begun [design work](#) with a goal of addressing this limitation.

Conclusion

Magma is designed to offer a scalable approach to building and operating a mobile core network. It supports 3GPP protocols at critical edge interfaces, specifically, the interface to the RAN (as shown in Figure 2), and the federation interface to other mobile networks (Figure 3). However, to achieve the benefits of an SDN-style approach, it does not use 3GPP protocols for internal communication between its components. The primary disadvantage of this approach is that a mobile core based on Magma cannot mix and match components from multiple vendors or open source projects. There are also limitations in today's implementation on the support for mobility across gateways. The advantages include higher performance and the ability to scale out the solution to thousands of base stations while maintaining a centralized point of control for ease of operation and configuration. Magma's approach also provides a solution that supports diverse types of backhaul links (e.g. satellite) with high resilience and graceful recovery from hardware and software failures.