

Bulut bilişimde sunucusuz mimariler ile coğrafi bilgi teknolojilerinin kullanımı üzerine bir inceleme

Mete Ercan Pakdil^{1*}, Rahmi Nurhan Çelik¹

¹İstanbul Teknik Üniversitesi, Ayazağa Kampüsü, İnşaat Fakültesi, Geomatik Mühendisliği Bölümü, Sarıyer, İstanbul, Türkiye.

Öz: Bulut bilişim, kaynakların yönetimi üzerinden üç farklı kategoriye ayrılmaktadır. Bu kategoriler Hizmet olarak Altyapı (Infrastructure as a Service, IaaS), Hizmet olarak Platform (Platform as a Service, PaaS), Hizmet olarak Yazılım (Software as a Service, SaaS) şeklinde isimlendirilmektedir. Her bir model, kullanıcı ve sağlayıcı üzerindeki altyapı ve uygulama yönetiminin sorumluluğunun paylaşılması ile açıklanabilir. Günümüzde bu modellere ek olarak sunucusuz mimari modeline dayanan Hizmet olarak Fonksiyon (Function as a Service, FaaS) kavramı, bu üç modelin sentezi olarak ortaya çıkmıştır. Bu da kısaca sunucusuz hizmetleri ifade etmektedir. Bu model, kısaca açıklanmak istenirse, geliştirilen kodun servis tarafından bir isteğe veya olaya bağlı olacak şekilde ölçeklenebilir olarak çalıştırılmasını sağlamaktadır. Kullanıcı sadece kodun geliştirilmesinden ve yüklenmesinden sorumludur. Giderek yaygınlaşan bu model sadece kod çalıştırmak için değil veri saklama ve veri analizi gibi servisler için de uygulanmıştır. Böylece modelin kapsamı genişletilmiştir. Bu çalışmada, farklı Coğrafi Bilgi Sistemleri (CBS) yazılım mimarileri ele alınarak sunucusuz mimarilerle bulut bilişim üzerindeki kullanımı incelenmiştir. Bu kapsamda çeşitli CBS yazılımları seçilmiş, bir bulut bilişim servis sağlayıcısının sağladığı bulut bilişim servisleri üzerinde kullanımları gerçekleştirilmiş ve elde edilen sonuçlar tartışılmıştır. Böylece giderek popülerleşen sunucusuz mimarilere dayalı servislerin ve yazılımların farklı CBS uygulamaları ile kullanımları incelenerek bir farkındalığın oluşması hedeflenmiştir. Tüm bunlara ek olarak CBS teknolojileri, mekânsal zekâ, büyük veri gibi bulut bilişim dünyasında her geçen gün popülerliği artan teknoloji ve oluşumların sunucusuz mimarilerle etkileşimi incelenmiş, geleceğin şekillendirilmesine katkı sağlayacak bir yol haritası verilmiştir.

Anahtar Sözcükler: Bulut bilişim, Coğrafi bilgi sistemleri, Sunucusuz mimariler

A review on the usage of geographic information technologies with serverless architectures in cloud computing

Abstract: From the perspective of resource management, cloud computing falls into three categories, namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each model can be clarified in the responsibility share of infrastructure and implementation management on the user and provider. Nowadays, as an addition to these models, the notion of Function as a Service (FaaS) has appeared, based on serverless architecture, implying serverless services. With this model, in brief, a developed code is run scalable depending on a request or an event. The user is responsible only for the development and uploading of the code. This increasingly widespread model has been applied not only to run the code but also to services such as data storage and data analysis. Thus, the scope of the model has been expanded. This study deals with various Geospatial Information Systems (GIS) from the viewpoint of their use on cloud computing with serverless architectures. In this context, various GIS applications were selected to deploy a public cloud computing provider's services, and results were discussed. In this way, it is aimed to raise awareness by examining the use of services and software based on increasingly popular serverless architectures with different GIS applications. In addition, the interaction of trending technologies and computing paradigms in the cloud computing area such as GIS technologies, spatial intelligence, and big data, with serverless architectures are reviewed; and a contributory road map for the future is proposed.

Keywords: Cloud computing, Geographic information systems, Serverless architectures

* Sorumlu Yazar/Corresponding Author: Tel: +44 793 103 4753

Geliş Tarihi/Received: 07.11.2021

Kabul Tarihi/Accepted: 08.03.2022



1. Giriş

Bulut bilişim teknolojisinin gelişimine paralel olarak, bu teknolojiden faydalanılarak yapılan çalışmalar bilim ve teknoloji dünyasında önemli gelişmelerin kaydedilmesini sağlamıştır. Özellikle büyük veriye dayalı, büyük hesaplama gücü gerektiren işlemler ve yüksek sayıda kullanıcıya sahip uygulamaların ihtiyacı olan altyapının bulut bilişim üzerinden kolay ölçeklenebilir ve düşük maliyetli bir şekilde sağlanabilmesi bu gelişmelerin önünü açmıştır. Bugün bulut bilişimin kullanılmadığı büyük veri senaryoları düşünülememektedir. Araştırmacılar geliştirdikleri algoritmaları hızlı bir şekilde bulut bilişim platformları üzerinde, istenen özelliklerde hazır olarak bulunan bilişim altyapılarını kullanarak test edebilmekte ve çalışmalarını daha hızlı sonuçlandırma olanağı elde etmektedirler. Bu da bilim ve teknolojinin ilerlemesine katkı sağlamaktadır. Bu teknoloji her alanda olduğu gibi mekânsal bilişim teknolojilerinin uygulanmasında ve hatta evrimleşmesinde de büyük rol oynamaya başlamıştır.

Mekânsal bilişim teknolojileri her geçen gün büyük veriyle çalışan analiz süreçlerinin giderek daha önemli bir parçası olmaya başlamış hatta büyük oranda ana oyuncusu haline gelmiştir (Dold & Groopman, 2017). Bu süreçlerdeki zorluk ise verinin büyüklüğünün yanında yapılmak istenen karmaşık analizlerin altından kalkacak bilgi teknolojilerinin alt yapısının kurulumu ve yönetimidir. Altyapı kurulumu ve yönetimi bulut bilişim sayesinde çok kolaylaşmış, hatta kimi durumlarda ihtiyaç olmaktan tamamen çıkmıştır. Bu nedenle günümüzde mekânsal bilişim hakkında çalışma yaparken artık bulut bilişimden bahsedilmemesi düşünülemez. Her ne kadar bir coğrafi bilgi teknolojisi projesi bu kavram üzerinden kurgulanarak gerçekleştirilmek istense de bu konuda çalışmak isteyen coğrafi bilgi teknolojileri uzmanlarının önündeki teknik zorluklar, çalışmayı yürütecek kurum için yeterli insan kaynağının bulunamaması ve ortaya çıkan maliyetler birer engel teşkil edebilmektedir.

Bulut bilişim konusunda yetişmiş insan kaynağının istihdam edilmesi, bu insan kaynağına olan talep artışı nedeniyle, bu teknolojiden faydalanmak isteyen kurumlar için bir engel olarak görülmektedir. Ayrıca aranan insan kaynağının coğrafi bilgi teknolojileri konusunda da yetkin olmasının beklenmesi ile istenen insan kaynağının bulunamadığı zorluk katlanarak artmaktadır.

Teknik zorluklar ise özellikle coğrafi bilgi teknolojileri konusunda bulut bilişim tabanlı bir proje geliştirilmek istendiğinde ayrıca değerlendirilmelidir. Bu teknik zorluklar çalışılacak verinin büyüklüğü ve projenin kullanıcı sayısı ile doğru orantılı olarak artmaktadır.

Bulut bilişimin temeli her ne kadar sınırsız donanım kaynağı ve ölçeklenebilirlik kavramları üzerine kurulu olsa da uygulamada sınırlı proje bütçeleri kaynakların kullanımlarını sınırlamaktadır. Bunun yanında birçok bulut bilişim servisi 'kullandıkça öde' fiyatlandırma modeli üzerine kurulu olduğu için eğer kaynaklar iyi yönetilemezse ve kontrolsüz bir şekilde büyüme gerçekleşirse proje yürütücüsü mali açıdan zor duruma düşecektir. Bahsedilen bu zorluklar sunucusuz bulut bilişim teknolojilerinin sunduğu olay bazlı çalışma ile kısmen veya tamamen aşılabilmektedir (Bebortta, Das, Kandpal, Barik, & Dubey, 2020). Örneğin devamlı çalışması gerekmeyen bir uygulamanın sürekli donanım kaynaklarını meşgul etmeyerek ve sadece istek zamanında donanım kaynaklarını tüketerek ücretlendirilmesi sağlanabilir.

Sunucusuz mimariler, bulut bilişim paradigması içinde günümüzde her geçen gün daha fazla tanınmaya ve kullanılmaya başlayan bir teknoloji olmuştur (Baldini vd., 2017). Özellikle bu teknolojik yaklaşımın diğer yaklaşımlara göre getirdiği en büyük fayda hesaplama sistemlerindeki yüksek ölçeklenebilirliği, yönetimsel teknik zorlukları kullanıcıdan soyutlayarak kolay ve bakım gerektirmeden sunmasıdır. Sunucusuz hesaplama servisleri Hizmet olarak Fonksiyon (Function as a Service, FaaS) ve Hizmet olarak Konteyner (Container as a Service, CaaS) olmak üzere iki ana türde incelenebilir (Chowhan, 2018;

URL-1). Bu iki servis türü de alt yapı yönetiminin büyük oranda servis kullanıcılarından soyutlanarak bilişim servis sağlayıcısı tarafından yönetilmesi yaklaşımıyla sunulur. Böylece hem altyapı yönetimindeki teknik zorluklar hem de altyapı yönetimi için gereken insan kaynağı büyük oranda azaltılmış olur. Bu teknolojinin Coğrafi Bilgi Sistemleri (CBS) uygulamalarında kullanılması ile bulut bilişim konusunda ileri düzeyde teknik yetkinliği olmayan CBS uzmanlarının bulut bilişimle tanışması da kolaylaştırılabilir.

FaaS mimarisiyle oluşturulan uygulamalar bir olayın tetiklemesi üzerine çalışırlar. Örneğin bu olay nesnelerin interneti teknolojisi ile çalışan akıllı bir duyargadan (sensör) gelebilir. Günümüzde çeşitli türden akıllı duyargalar aynı zamanda akıllı şehir ekosistemlerinin önemli bir oyuncusu olup yaygın olarak kullanılmaktadır. Bu duyargalardan gelen verilerin çevresel olgu ve faaliyetlere dayalı olarak ürettiği verilerin gerçek zamana mümkün olduğunca yakın olarak işlenebilmesi karar vericiler için önemli olmaktadır. Bu durumda verinin işleneceği CBS alt yapısının da hızlı cevap veren ve kolay ölçeklenebilir bir şekilde olması gerekmektedir. Bu sayede gelen verilerin en hızlı şekilde saklanması ve analizi sağlanabilmektedir (Evans vd., 2019). Örnek olarak, akıllı duyargalardan gelen verilerin işlendiği bir akıllı şehir uygulaması senaryosunda, bulut bilişim üzerindeki diğer hizmet türleri olan Hizmet olarak Altyapı (Infrastructure as a Service, IaaS) ve Hizmet olarak Platform (Platform as a Service, PaaS) üzerine kurulu sistem mimarilerinin başarısı ve sürdürülebilirliği, alt yapının yönetimi bu hizmetlerin kullanıcılarına bırakıldığı için teknik yetkinlik ve insan kaynağına bağlı olacaktır. Sunucusuz mimariye dayalı hizmetlerin kullanımında ise bu senaryonun başarısı sadece kullanılan veya geliştirilen uygulamanın yeteneklerine ve verimliliğine bağlı olmaktadır (van Eyk vd., 2018). Böylece sadece uygulamanın doğru optimizasyonu ve bakımı ile sistemin devamlılığı sağlanabilir ve verimliliği artırılabilir.

Sunucusuz mimarilere dayalı bilişim teknolojilerinin ilk bakışta her ne kadar bulut bilişim sağlayıcılarının sunduğu altyapı olmadan çalışamayacağı düşünülse de aslında kurum içindeki bir mevcut altyapı üzerinde de çalıştırılabilir. Birçok açık kaynak kodlu sistem yönetimi yazılımı sayesinde kurum içindeki sunuculara da sunucusuz mimariyle çalışan hizmetlerle bir sistem kurulabilir. Böylece bulut bilişimin kullanılmasının mümkün olmadığı coğrafi bilgi teknolojileri projelerinin de kurum içi bilişim teknolojileri alt yapıları ile çalışması sağlanmaktadır. Bu sayede sunucusuz mimarinin en büyük kazanımlarından biri olan kaynak optimizasyonu kurum içindeki sistemlerde de uygulanabilir. Ancak bu kurulumun da bilişim teknolojileri konusunda yetkin insan kaynağı tarafından yapılması gerekeceği düşünülürse bu konudaki insan kaynağı ihtiyacı ortadan kalkmayacaktır.

Günümüzde akıllı şehir uygulamalarında kullanılan mekânsal bilgi işleme süreçleri, harita servisleri, belli aralıklarla çalışan mekânsal analizleri gibi CBS uygulamaları devamlı donanım kaynağı tüketmesi gerekmez ve sadece planlanan zamanda veya istek anında çalışmak üzere beklerler. Yukarıda bahsedilen sunucusuz mimarilerin getirdiği avantajlar göz önüne alındığında bu uygulamaların sunucusuz mimarilerle tasarlanması sistem kaynaklarının daha verimli kullanılması ile optimizasyonunu sağlar. Ayrıca alt yapı bakımının da soyutlanması ile daha yönetilebilir ve kolay ölçeklenebilir bir sistem kurulması mümkün hale gelmektedir.

Bu çalışmada mevcut CBS uygulamalarının sunucusuz mimariye dayalı teknolojilerle ve seçilen bir bulut bilişim hizmeti sağlayıcısına ait sunucusuz servislerle tasarımları sunulmuştur. Böylece mevcut uygulamaların sunucusuz mimarilerle kullanımları incelenmiş ve teknik zorlukları ile avantajları irdelenmiştir. Çalışmada, sistem tasarımlarında kullanılan temel sunucusuz bulut bilişim hizmet türleri irdelenmiştir. Verilen örnekler bir bulut bilişim sağlayıcısının servisleri seçilerek irdelenmiş olsa da benzer servisler diğer bulut bilişim sağlayıcılarında da bulunarak sunulan sistem tasarımları uygulanabilir.

2. İlgili Çalışmalar

Baldini vd. (2017) sundukları çalışmada sunucusuz mimarileri tanımlamış ve günümüzdeki kullanım alanlarını incelemiştir. Bununla birlikte bu çalışmada mevcut sunucusuz mimariye dayalı platformların karakteristik özellikleri tanımlanarak yeni ve gelecekte yayınlanacak açık kaynak sunucusuz mimariye dayalı yazılımlardan da bahsedilmiştir. En sonunda sunucusuz mimari yaklaşımının teknik zorlukları, dezavantajları ve avantajları ile birlikte açık araştırma konularına da değinilmiştir.

Bebortta vd. (2020) yaptıkları çalışmada sunucusuz mimarilerle coğrafi büyük verinin analizini farklı sunucusuz bilişim servis sağlayıcısı platformlar ve yazılımla inceleyerek uygulanabilirliğini değerlendirmişlerdir. Ayrıca çalışmada sundukları örnek bir coğrafi veri analizi çerçevesinin sunucusuz bilişim paradigması ile uygulaması yapılarak değerlendirilmiştir.

Anand, Johnson, Mathiksara ve Kartik (2019) yayınladıkları çalışmada geliştirdikleri gerçek zamanlı konum takip sistemini GPS tabanlı donanım ve sunucusuz servislerle birlikte incelemiştir. Çalışmada GPS alıcısından alınan konumların saklanması için bir bulut bilişim platformu üzerinde sunucusuz veri depolama hizmeti kullanılmıştır.

Mete ve Yomralıoğlu (2021a) yaptıkları çalışmada bir arazi değerlemesi uygulamasının verilerinin sunucusuz bulut bilişim servisleri kullanılarak yayınlanması için bir yazılım mimarisi kurgulamış ve uygulamışlardır. Bu çalışma yazarların daha önce sundukları Mete ve Yomralıoğlu, (2021b) çalışmasındaki bulut bilişim tabanlı yazılım mimarisinin vektör karo sunucusuz servislerle yeniden tasarlanarak sunulmasını göstermektedir. Bu bağlamda mevcut coğrafi bilgi sistemi uygulamalarının sunucusuz servislere taşınması için literatürde örnek olmaktadır.

Pakdil ve Çelik (2021) yayınladıkları çalışmada sunucusuz mimarilerle mekânsal veri işleme iş akışları çalıştırabilecek bir sistem tasarımı sunmuştur. Bu tasarımda veri işleme ve web servislerin yayını için hem FaaS hem de CaaS türündeki servisler birlikte kullanılmıştır. Ayrıca çalışmada OGC (Open Geospatial Consortium) API (Application Programming Interface) Features tanımının sunucusuz mimarilerle çalışabilirliği de incelenmiştir.

3. Sunucusuz Mimariye Dayalı Mevcut Teknolojilere Genel Bakış

Sunucusuz mimariye dayalı teknolojiler sunucu yönetimi gerekmeyen sistemler olarak tanımlanır. Sunucusuz mimari sunucu olmadığı anlamına gelmez ancak çalışacak yazılım için önceden sunucu kaynağı ayrılması, ölçeklendirilmesi ve bakımı gerekmez (URL-2). Literatür incelendiğinde sunucusuz mimariye dayalı teknolojiler Tablo 1'de görüldüğü gibi veri depolama ve hesaplama olarak iki ana çatı altında toplanabilirler. Veri depolama türündeki hizmetler de yapısal ve yapısal olmayan veri türlerine göre iki alt gruba ayrılabilir (URL-3). Benzer şekilde hesaplama türündeki hizmetler de FaaS ve konteyner CaaS tabanlı alt yapı türüne göre iki grupta incelenmektedir (Chowhan 2018; URL-1).

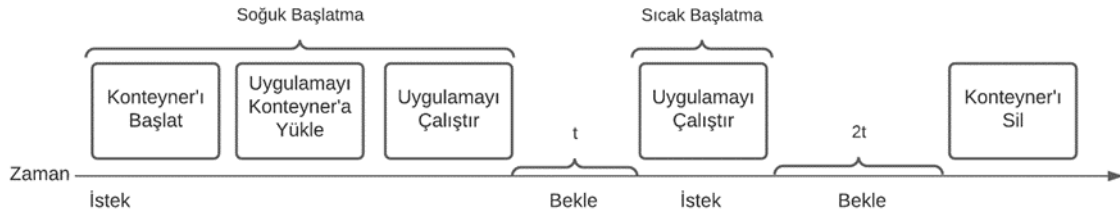
Tablo 1: Sunucusuz mimariye dayalı teknoloji türlerinin sınıflandırılması

Veri Depolama Hizmeti	Hesaplama Hizmeti
Yapısal ve ilişkisel veri saklama	FaaS
Yapısal olmayan veri saklama	CaaS

Yapısal ve ilişkisel veri saklama hizmetleri genel olarak Yapısal Sorgulama Dili'ni (Structured Query Language, SQL) destekleyen yazılımlar veya doküman veri tiplerini saklayan ve internet servisleri ile sorgulatan yazılımlardan oluşmaktadır. Bu yapısal ve ilişkisel veri saklama servislerinin bazıları açık kaynak kodlu yazılımlar aracılığıyla sunulmuş olup bazıları da bulut bilişim sağlayıcı firmanın özel olarak ürettiği yazılımlar sayesinde sunulmaktadır. Yapısal ve ilişkisel veri saklama

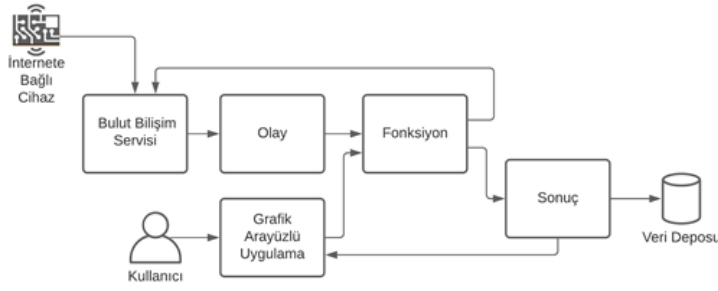
hizmetlerinin birçoğu coğrafi veri eklentileri veya dâhili coğrafi veri işlemcileriyle mekânsal sorgulara ve veri saklanmasına destek vermektedirler. Yapısal olmayan veriler İkili Büyük Nesne (Binary Long Objects, BLOB) türünde saklanmaktadır. İkili biçimde olan her şey BLOB olarak saklanabilir (Amirian, Basiri, & Winstanley, 2014). En yaygın biçimler; resim, Virgülle Ayrılmış Değerler (Comma Separated Values, CSV), Apache Parquet, düz yazı gibi biçimlerdir. Yapısal olmayan veri saklama hizmetleri üzerinde saklanan coğrafi verilerin sorgulanması için bir başka sunucusuz sorgulama servisi gerekebilir. Yapısal olmayan verilerin mekânsal indekslenmesi uygulama geliştirme arayüzleri üzerinden verilere hızlıca erişerek sorgulanmasını mümkün kılmaktadır (URL-4). Yapısal olmayan veriler sunulan internet servisleri ile yönetilebilirler. Ayrıca yapısal olmayan veriler üzerinde veya saklama alanındaki değişiklikler birer olay üretir ve FaaS'i tetikleyerek çalıştırabilir. Böylece istenirse gerçek zamana yakın bir şekilde değişen veriler üzerinde analiz yapılabilir veya bir başka servisin değişen veri hakkında bilgi üretmesi sağlanabilir.

Sunucusuz hesaplama servislerinden biri olan fonksiyonlar, belli bir amaç için geliştirilen yazılım parçasını çalıştırmak için kullanılırlar. Bu yazılım parçası kullanılacak alt yapının desteklediği herhangi bir programlama dilinde geliştirilebilir. Fonksiyonlar sadece istek anında çalışmaya başlarlar. Temelde birer konteyner yönetimi olarak çalışan FaaS mimarileri, özel olarak geliştirilen konteynerlerin sisteme yüklenen kod parçalarını istek alındığı anda olabilecek en hızlı şekilde çalıştırılması üzerine tasarlanmıştır. Şekil 1, Chowhan (2018) çalışmasından yararlanılarak çalışma döngüsünü açıklamak için oluşturulmuştur.



Şekil 1: Bir fonksiyonun çalışma döngüsü

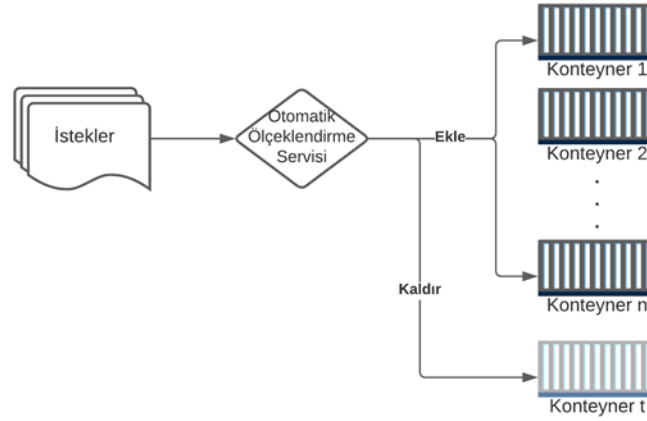
İstemci yazılım fonksiyona senkron bir istek gönderdikten sonra, sistem fonksiyon konfigürasyonunda belirtilen programlama diline uygun konteyneri çalıştırmaya başlar. Bu aşamadan sonraki geçen sürede istemci yazılım fonksiyonundan yanıt beklemeye başlar. Bu sürece soğuk başlatma süreci denir. Çalışmaya başlayan fonksiyon uygulamasının istemci yazılıma yanıt vermesiyle soğuk başlatma süreci tamamlanmış olur. Eğer bir fonksiyon uzun süre belli aralıklarla istek almaya devam ederse soğuk başlatma sürecinde çalıştırılan konteyner kapatılmadan çalışmaya devam eder. Buna sıcak başlatma denir. Bu nedenle bir fonksiyonun ilk isteğe verilen yanıt süresi ardı sıra gelen isteklere göre daha uzun olmaktadır. Fonksiyonlar kolaylıkla ölçeklenebilirler. Bir fonksiyon, çalıştığı konteynerin hesaplama kapasitesinin üzerinde istek almaya başlarsa orkestrasyon yazılımı otomatik olarak yeni bir konteyner çalıştırarak gelen istekleri eşit olarak dağıtır ve işlem yükünü bölüştürmüştür. Böylelikle fonksiyonlar için geliştirilen kodların ek olarak elle bir müdahale gerekmeden çalışmaya başlaması beklenir. Aksi halde orkestrasyon yazılımı gerekli otomasyonu sağlayamayacaktır. Kaynak tüketimi ve bununla orantılı olarak maliyetlerin artmasına sebep olacağından, fonksiyonların kısa zamanda işlemleri bitirmesi gerekir (Adzic & Chatley, 2017). Fonksiyonların çalışma süresi sonunda ürettikleri sonuçlar başka bir bulut bilişim servisini tetikleyerek başka bir servise aktarılabilir ya da bir Hiper-Metin Transfer Protokolü (Hyper-Text Transfer Protocol, HTTP) ile internet servisi olarak sunulabilir. Barbieri ve Bonanni (2019) çalışmasından yararlanılarak fonksiyonun diğer servislerle olan ilişkisini açıklamak üzere Şekil 2 oluşturulmuştur.



Şekil 2: Bir fonksiyonun diğer uygulama ve servislerle ilişkisi

Sunucusuz CaaS servisleri fonksiyon servislerine göre daha özgür çalışma ortamı sunar. Bunun nedeni ise uygulama geliştirici için fonksiyonların aksine sadece kodun üzerinde değil çalışılan konteynerin üzerinde de yetki alanı sunulmasıdır (URL-5). Böylece bir uygulama geliştirici geliştirdiği uygulamanın hangi ortamda ve hangi uygulama sunucusu yazılımı ile çalışabileceğine karar verebilir. Bu bağlamda, mevcut CBS uygulamalarını çalıştırmak için de konteyner teknolojisi kullanılabilir (Zaragozí, Trilles, & Navarro-Carrión, 2020). Örneğin MapServer isimli açık kaynak kodlu CBS uygulaması Docker konteyner formatında sunulmaktadır (URL-6). Literatür incelemesi yapıldığında birçok farklı konteyner platformu olduğu görülmektedir (Siddiqui, Siddiqui, & Khan, 2019). Bu platformlar içerisinde Docker isimli platform birçok farklı bulut bilişim sağlayıcısı tarafından desteklendiği için öne çıkmaktadır. Docker platformu üzerinde hazırlanan konteynerler birer imaj olarak bir depoda tutulurlar (Nickoloff & Kuenzli, 2019; URL-7). CaaS ilgili imajı depodan indirir ve konteyneri çalıştırır. Konteynerler devamlı olarak çalışmaya devam edebileceği gibi sadece bir görevi tamamlamak üzere de çalışıp durabilirler. Devamlı çalışan konteynerlere örnek olarak internet sunucusu uygulamaları gösterilebilir. Bu uygulamalar bir internet iletişim protokolü üzerinden gelecek olan isteği devamlı olarak beklerler. Eğer aynı anda gelen istek sayısı artarsa CaaS gelen talebi karşılamak üzere çalışan konteyner sayısını artırabilir. Bu nedenle konteynerler de fonksiyon uygulamalarına benzer şekilde orkestrasyon yazılımı tarafından otomatik olarak başlatılıp elle müdahale gerekmeden istek almaya hazır olmalıdırlar. Eğer sistem üzerindeki çalışma yükü azalır CaaS ihtiyaç duymadığı konteynerleri otomatik olarak kapatır.

Konteynerlerdeki veri konteyner kapatıldığında silindiğinden dolayı çalışan uygulamalar konteyner üzerinde veri saklamamalıdırlar. CaaS üzerinde çalışan internet uygulamaları genel olarak bir yük dengeleyici servisi ile çalışırlar. Otomatik ölçeklendirme servisi yeni konteynerlerin eklenmesi veya gereksiz konteynerlerin silinmesi emrini izlediği metriklere göre karar vererek orkestrasyon yazılımına iletir (Tang & Wang, 2020). Bu sayede kaynak tüketiminin optimizasyonu sağlanır ve bir konteyner filosu otomatik olarak yönetilmiş olur Şekil 3, Luksa (2017) çalışmasından yararlanılarak otomatik ölçeklendirme servisini açıklamak için oluşturulmuştur. Konteyner mimarileri tek bir konteyner imajının ölçeklenerek çalıştırılması şeklinde olabileceği gibi birden fazla konteyner imajının birbirleri ile iletişim kurarak çalıştırıldığı bir sistem olarak da tasarlanabilir. Bu sistemlerde ağ yönetimi soyutlanarak CaaS tarafından gerçekleştirilir.



Şekil 3: Otomatik ölçeklendirme servisi ile konteyner yönetimi

Bir sonraki bölümde bulut bilişim sağlayıcılarından olan Amazon Web Services (AWS) firmasının sunucusuz olarak sağladığı veri saklama ve hesaplama hizmetlerinin mekânsal bilişim teknolojileri alanındaki kullanımı, ücretlendirme şekilleri ve mevcut çalışma kısıtları incelenecektir.

3.1 Amazon Web Services (AWS) Bulut Bilişim Sağlayıcısının Sunduğu Sunucusuz Hizmetler

AWS firmasının yapısal olmayan veri depolama için sunduğu sunucusuz hizmeti Amazon S3 olarak isimlendirilmektedir. Amazon S3 hizmeti verileri BLOB türünde saklamaktadır. Bu hizmet saklanan verinin ve saklama alanına yapılan veri trafiğinin büyüklüğüne bağlı olarak ücretlendirme politikası izlemektedir. Amazon S3 üzerinde saklanan CSV, Apache Parquet veya düz metin gibi biçemlerdeki dosyaların içerdikleri coğrafi verilerin sorgulanması için bir başka sunucusuz sorgulama servisi Amazon Athena kullanılabilir.

Yapısal verilerin saklanması için AWS firması tarafından veri saklama hizmetleri olarak Amazon Aurora Serverless ve Amazon DynamoDB servisi sunulmaktadır. Bu servis açık kaynak kodlu PostgreSQL ve MySQL veri tabanı sunucusu yazılımlarının AWS tarafından özelleştirilmiş bir sürümleri üzerinden sunulmaktadır. Bu servisin ücretlendirilmesi saatlik aktif olan veri tabanı sunucusunun sayısı, saklanan veri miktarı ve yapılan sorgulama sayısı üzerinden yapılmaktadır. Örneğin, saklanan veriler hiçbir zaman sorgulanmazsa sadece depolama ücreti ödenmektedir. Bir diğer örnek ise; verilerin ayda bir kez bir saat boyunca sorgulanması durumunda, yapılan sorgunun kullandığı sunucuların bir saatlik kullanım ücreti ve saklanan veri miktarı üzerinden aylık olarak ücretlendirilir. PostgreSQL eklenti ekosisteminde mekânsal verinin saklanması ve analizi için geliştirilmiş olan PostGIS eklentisi Amazon Aurora Serverless ile çalışabilmektedir. PostGIS eklentisi ile birçok farklı türden mekânsal veri saklanabilir ve SQL sorguları ile gelişmiş analizler yapılabilir. Bu sayede mekânsal verilerin saklanması ve ileri düzeyde sorgulanması mümkün olmaktadır (Mete & Yomrahoğlu, 2021a).

Amazon DynamoDB anahtar-değer doküman türü verileri saklama hizmetidir. Amazon DynamoDB'nin ücretlendirilmesi ise veri okuma ve yazma üzerinden kullanım miktarına göre yapılmaktadır. Bu hizmet coğrafi verilerin saklanmasına ve sorgulanmasına kısıtlı olarak destek vermektedir; sadece nokta türündeki coğrafi verilerin GeoHash kodlaması ile saklanmasını tavsiye etmektedir. GeoHash kütüphaneleri ile uygulama üzerinden coğrafi veriler kısıtlı olarak analiz edilebilmektedir (URL-8). Amazon DynamoDB veri sorgulama işlemlerini SQL dili desteğini kısıtlı bir şekilde vererek ve kendine özgü sorgulama fonksiyonlarını internet servisleri üzerinden sağlayarak sunmaktadır.

AWS firmasının sunduğu sunucusuz hesaplama hizmetleri FaaS ve CaaS türünde ikiye ayrılmaktadır. FaaS türündeki servisin adı AWS Lambda olarak adlandırılmaktadır (Chapin & Roberts, 2020). AWS Lambda fonksiyonları çalışma süresi boyunca

atanan bellek miktarına ve toplam çalışma süresine göre ücretlendirilirler. AWS Lambda fonksiyonlarının bellek kapasitesi kullanıcı tarafından belirlenir. Bu yüzden uygulamanın ihtiyacı olan bellek miktarının doğru tespitini yapmak ve gereğinden fazla bellek kapasitesi atamamak önem arz etmektedir. AWS Lambda üzerindeki fonksiyonlar diğer AWS servisleriyle iletişim kurabilirler. Örneğin, AWS Lambda üzerinde Amazon DynamoDB kullanan bir veri tabanı uygulaması da çalıştırılabilir. AWS Lambda servisinin çalışma süresi 15 dakika ile limitli olduğundan, çalışması uzun süren mekânsal büyük veri analizi veya mekânsal zekâ uygulamaları için uygun olmayabilir. Bununla birlikte bir fonksiyonun aynı anda en fazla bin isteğe cevap verebilme limiti bulunmaktadır. Özellikle herkese açık ve çok sayıda kullanıcıya hizmet vermesi beklenen internet servisi mimarileri için bu limit önemlidir. Servis sağlayıcı firmadan talep edilerek bu limit arttırılabilmektedir. AWS Lambda servisi başlıca Java Script, Python, Ruby, Java, Go ve C# programlama dillerini desteklemektedir, ancak, servis ayrıca konteyner imajı çalıştırma desteği verdiği için uygulamada konteyner üzerinde çalışabilen her dille çalışma imkânı da sunmaktadır. AWS Lambda servisinin konteyner çalıştırma yeteneği açısından CaaS modeline dayalı AWS Fargate servisi ile ortak özellikler taşıdığı görülmektedir.

AWS firmasının CaaS türündeki servisi AWS Fargate olarak adlandırılmaktadır. Bu hizmetin ücretlendirilmesi işlemci sayısının ve bellek miktarının kullanım sürelerine göre hesaplanmaktadır (Vohra, 2018). Bununla birlikte konteyner başına geçici olarak 20 GB depolama alanı ücretsiz olarak sunulmuştur; ancak daha büyük bir depolama kapasitesi atanırsa bunun da ücretlendirilmesi kullanılan kapasiteye ve süresine bağlı olarak değişmektedir. Eğer kalıcı bir depolama ihtiyacı varsa bunun için sunucusuz yapısal olmayan veri depolama hizmetleri kullanılabilir gibi disk hizmetinden de faydalanılabilir. Servis aynı anda bin adet konteyner çalıştırabilir; ancak bu limit servis sağlayıcı firmadan talep edilerek arttırılabilmektedir. Bir diğer limit ise AWS Fargate servisinin grafik işlem birimi desteği sunmamasıdır. Bu nedenle grafik işlem birimi ile çalışan mekânsal zekâ uygulamaları AWS Fargate üzerinde çalışmamaktadır.

Bir sonraki bölümde AWS firmasının sağladığı Tablo 2’de belirtilen ve yukarıda açıklanan sunucusuz veri depolama ve hesaplama servisleri kullanılarak çeşitli mekânsal bilişim uygulamalarının mimari tasarımları sunulmaktadır.

Tablo 2: AWS firmasının sağladığı sunucusuz hizmetlerin sınıflandırılması

		Hizmet Adı
Veri Depolama Türü	Yapısal ve ilişkisel veri saklama	Amazon Aurora Serverless
	Yapısal olmayan veri saklama	Amazon S3 Amazon DynamoDB
Hesaplama Türü	Fonksiyon (FaaS)	AWS Lambda
	Konteyner (CaaS)	AWS Fargate

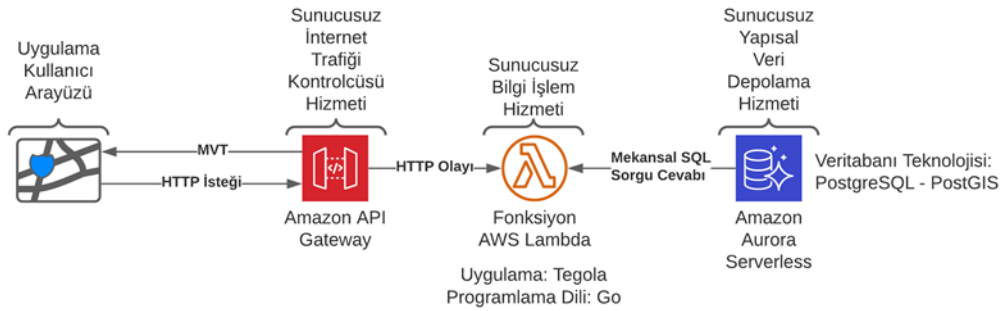
4. Sunucusuz Mimariye Dayalı Mekânsal Bilişim Uygulamalarının İncelenmesi

Bu bölümde sunucusuz mimariye dayalı dört farklı mekânsal bilişim servisi uygulaması incelenecektir, bu uygulamalar bir önceki bölümde incelenen AWS firmasının sağladığı sunucusuz hizmetler üzerinde çalışacak şekilde tasarlanmıştır. Aynı mimari tasarımlar diğer bulut bilişim servis sağlayıcılarında da veya yerel ağda tasarlanacak bir sunucusuz mimariye dayalı altyapı üzerinde de uygulanabilir.

4.1 Sunucusuz Vektör Karo Harita Servisi Tasarımı

Günümüzde birçok harita servisi vektör karo modelini kullanarak herkese açık yayın yapmaktadır. Bu servislerin çalışma prensibi, temelde, bir istemci uygulaması aracılığı ile mekânsal bir alanın karolara bölünerek sorgulanmasına dayanır. Her bir karonun satır ve sütun numaraları ile detay seviyesi sorgu parametresi olarak harita sunucusuna gönderilir. Harita sunucusu bu parametrelerle keşiften coğrafi alana ait nesnelere özel olarak kodlayarak istemci uygulamasına cevap olarak geri

gönderir ve istemci uygulama cevabı harita olarak görselleştirir (Wan, Huang, & Peng, 2016). İstemci uygulamanın bir vektör karo harita internet servisi ile iletişim kurabilmesi için bu servise ait iletişim kurallarını bilmesi gerekir. Vektör karo harita servisi tasarımında Mapbox Vector Tile (MVT) tanımlaması seçilmiştir (URL-9). Bu tanımlamaya göre vektör karo sorgu istekleri harita sunucusuna HTTP protokolü üzerinden HTTP GET metodu kullanılarak karo parametreleri gönderilmektedir. Harita sunucusu aldığı bu parametrelerle kesişen mekânsal veriyi veri tabanı sunucusundan sorgulayarak ister. Veri tabanından gelen veri Google Protobuf veri değişim formatı ile kodlanarak istemci uygulamaya HTTP cevabı olarak geri döndürülür.

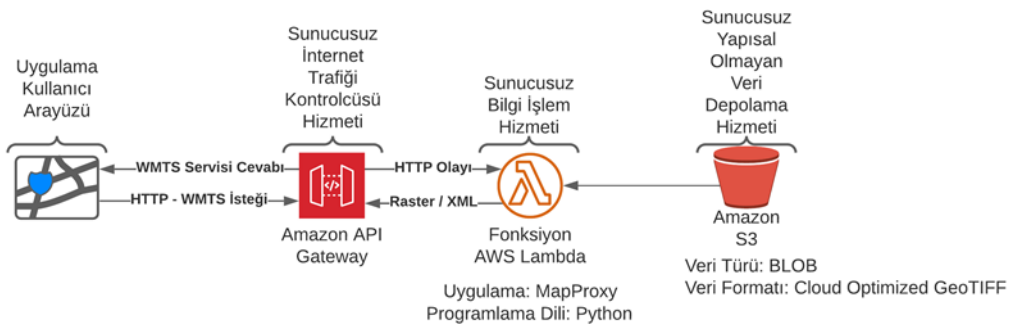


Şekil 4: Sunucusuz vektör karo harita servisi tasarımı ve bileşenleri

Şekil 4’te belirtilen mimari tasarımda FaaS üzerinde çalışmak üzere açık kaynak kodlu Tegola isimli harita sunucusu yazılımı kullanılmıştır. Bu yazılım Go programlama dilinde geliştirilmiştir ve MVT tanımlamasına göre çalışmaktadır (URL-10). Tegola yazılımı mekânsal veri tabanı olarak PostgreSQL üzerinde PostGIS eklentisine destek vermektedir. Bu sebeple sunucusuz veri depolama çözümü olarak Amazon Aurora Serverless adlı sunucusuz veri tabanı hizmeti bu mimari tasarımında tercih edilmiştir. FaaS’in bir olayla tetiklenmesi için Amazon API Gateway isimli internet trafiği kontrol ve yönlendiricisi olarak görev yapan sunucusuz uygulama servisine de ihtiyaç duyulmuştur. Bu servis gelen HTTP isteklerinin birer tetikleyici olaya dönüştürülerek AWS Lambda fonksiyonuna iletilmesinde ve fonksiyon ile istemciler arasındaki trafiğin sağlıklı bir şekilde yönetilmesinde kullanılmaktadır (Patterson, 2019).

4.2 Sunucusuz Raster Karo Harita Servisi Tasarımı

Harita sunucusu yazılımlar, bir OGC standardı olan Web Map Tile Service (WMTS) ile bir mekânsal alana ait daha önceden üretilmiş raster karolarını harita servisi olarak sunarlar (Gratier, Spencer, & Hazzard, 2015). Bu raster karoları farklı resim dosya biçimlerinde üretilebilirler. Bu çalışmada raster karoların dosya biçimi için Cloud Optimized GeoTIFF biçimi seçilmiştir. Bu biçim sayesinde veri sunucusu ile HTTP protokolü üzerinden olan iletişim normal GeoTIFF dosyalarına kıyasla daha hızlı olmaktadır (Flasher, 2019; URL-11).



Şekil 5: Sunucusuz raster karo harita servisi tasarımı ve bileşenleri

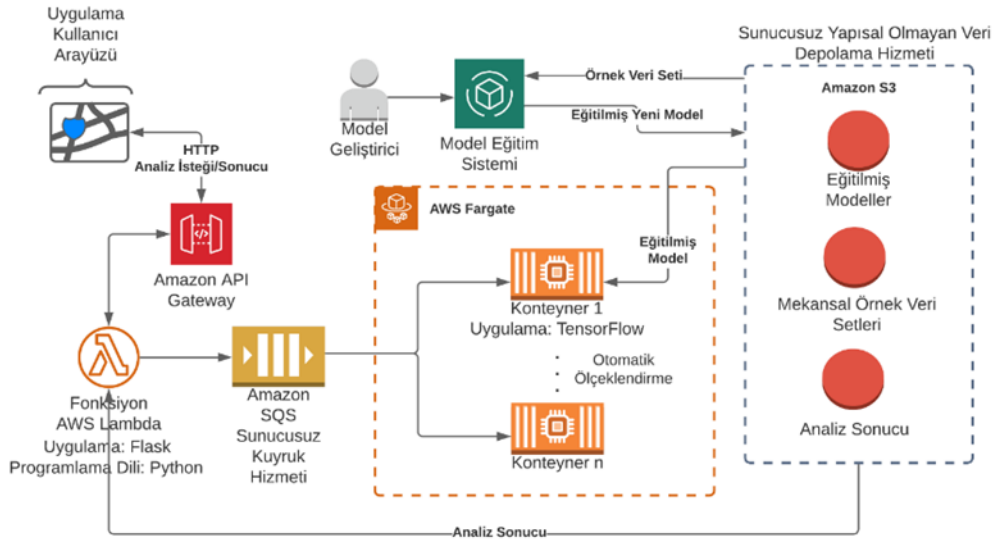
Şekil 5'te belirtilen mimari tasarımda önceden üretilen raster karo dosyaları, yapısal olmayan veri olarak Amazon S3 hizmetinde saklanacak şekilde konumlandırılmıştır. Sunucusuz raster karo harita servisi tasarımında FaaS üzerinde çalışmak üzere MapProxy isimli açık kaynak kodlu harita sunucusu yazılımı seçilmiştir (URL-12). Bu yazılım Python programlama dilinde geliştirilmiştir. MapProxy yazılımı ayrıca Web Map Service (WMS), WMTS, Keyhole Markup Language (KML) ve Tile Map Service (TMS) gibi harita yayın standartlarını destekleyen bir harita sunucusu yazılımıdır (Bateira, 2012). MapProxy yazılımı, raster karo dosyalarının Amazon S3 üzerinden okunmasına destek vermektedir. AWS Lambda fonksiyonu üzerinde çalışan MapProxy yazılımı bu tasarımda WMTS kullanıcı arayüzü uygulaması üzerinden gelen isteklere cevap vermektedir. Bununla birlikte WMTS standardını destekleyen diğer harita istemcisi uygulamalar da bu sistemden faydalanabilirler. Önerilen tasarımda FaaS'in harita istemcisi uygulamalarla HTTP protokolü üzerinden iletişim kurabilmesi için Amazon API Gateway isimli internet trafiği kontrol ve yönlendiricisi olarak görev yapan sunucusuz uygulama servisi kullanılmıştır.

4.3 Sunucusuz Mekânsal Veri Analizi Servisi Tasarımı

Mekânsal zekâ, mekânsal veri analizi uygulamalarında kullanımı giderek önem kazanan bir araştırma konusu haline gelmiştir. Güney ve Çelik (2020)'ye göre mekânsal zekâ şu şekilde tanımlanmıştır:

“Dünya üzerinde açık alanda ve/veya kapalı alanda gerçekleşen mekânsal veriyle ilişkili olguların ve insan faaliyetlerinin mekânsal modellemesinde, analizinde, görselleştirilmesinde, mekânsal problemlerin çözümünde ve mekânsal karar vermede yapay zeka yöntemlerinin yukarıda ifade edilen ileri teknolojilerle birlikte toplum yararına kullanımı mekânsal zeka olarak tanımlanabilir”.

Bu çalışmada yapay zeka ve makine öğrenmesi konularında yaygın olarak kullanılan TensorFlow isimli açık kaynak kodlu yazılım kullanılarak mekânsal veri analizi servisi tasarımı yapılmıştır (Audevart, Banachewicz, & Massaron, 2021). Önerilen tasarımda ayrıca beş farklı sunucusuz bulut bilişim hizmeti kullanılmıştır (Şekil 6).



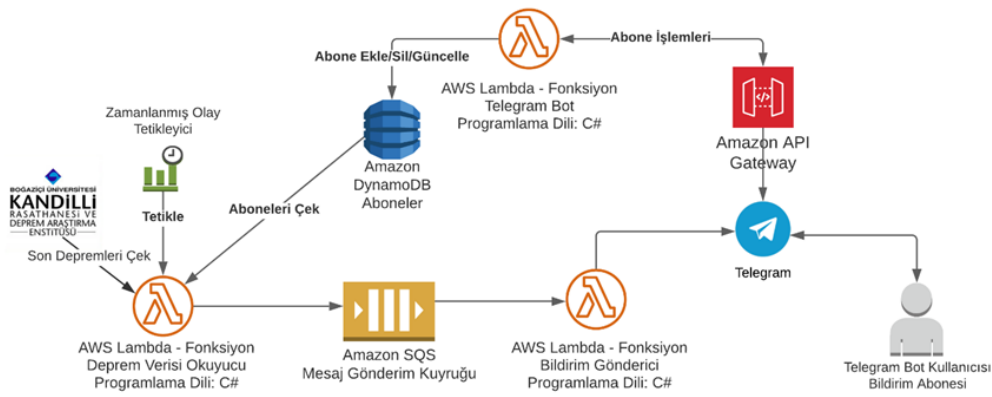
Şekil 6: Sunucusuz mekânsal veri analizi servisi tasarımı ve bileşenleri

Bu mimaride iki türlü rol bulunmaktadır. Uygulama kullanıcısı, uygulama kullanıcı arayüzünü kullanarak gerçekleştirmek istediği mekânsal analiz için istek göndermektedir. Model geliştiricisi ise analiz için kullanılan modeli yeni veri setleriyle veya algoritmalarla geliştirmekte ve eğitmektedir. Model geliştiricisi geliştirdiği yeni modeli sunucusuz mimariyle çalışan yapısal olmayan veri depolama hizmetine yükler. Yüklenen yeni model bir sonraki mekânsal analizde çalışacak konteyner

tarafından indirilerek kullanılacaktır. Uygulama kullanıcı arayüzünden HTTP protokolü ile gelen her yeni analiz isteği FaaS üzerinde çalışan bir HTTP servisi tarafından yine sunucusuz olarak çalışan kuyruk servisine eklenir. İsteği gönderen uygulama kullanıcıya işin sıraya konduğu bilgisi geri gönderilir. Otomatik ölçeklendirme servisi konteyner sayısını kuyrukta bekleyen işlem sayısına göre artırıp azaltabilir. Eğer kuyrukta bekleyen bir analiz işi yoksa çalışan konteyner sayısı sıfır olacaktır. İçerisinde TensorFlow uygulaması olan konteyner imajı CaaS tarafından bir konteyner imaj deposundan çekilerek konteynerler gerekli sayıda çalıştırılır. Konteyner çalışmaya başladıktan sonra eğitilmiş yeni modeli çeker ve kuyrukta bekleyen sıradaki analiz isteğini işleyerek oluşan analiz sonucunu sunucusuz yapısal olmayan veri depolama hizmetine yükler. Uygulama kullanıcıya üretilen mekânsal veri analizinin bittiği bilgisini alarak görüntüler. Bu mimarideki TensorFlow uygulaması, AWS Fargate servisi grafik işlem biriminin kullanımına izin vermediği için merkezi işlem birimi üzerinden analizleri çalıştırmaktadır (URL-13). Model eğitim sistemi bu tasarımın bir parçası değildir ancak sürecin nasıl çalıştığının daha iyi anlaşılması ve model geliştiricisi rolünün uygun şekilde temsili için tasarım üzerinde belirtilmiştir.

4.4 Sunucusuz Olay Tabanlı Mekânsal Veri İşleme Servisi Tasarımı: Kandilli Deprem Habercisi Örneği

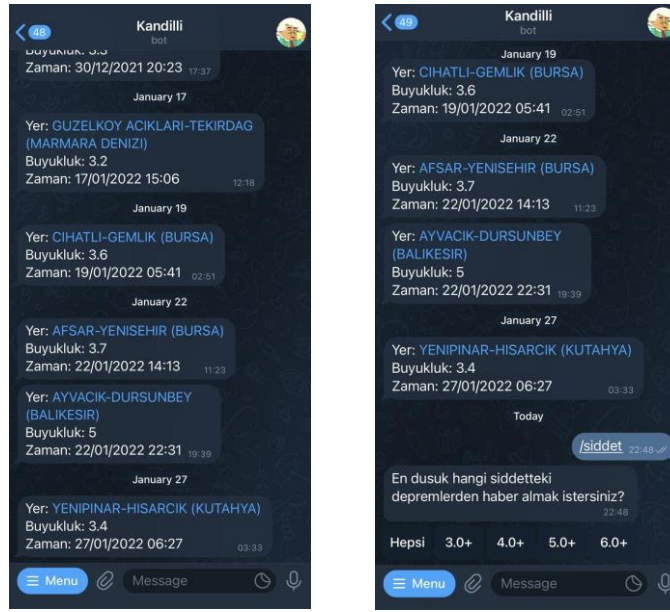
Bu bölümde olay kaynaklı çalışan bir sunucusuz mimariye dayalı mekânsal veri işleme uygulamasının tasarımı ve örneği incelenecektir. Bu tasarımda birden fazla FaaS bir arada farklı görevlerde çalışmaktadır (Şekil 7). Bu çalışmadaki amaç; sunucusuz bulut bilişim hizmetleri kullanılarak abonelerin Türkiye’de gerçekleşen son depremlerden kendi belirledikleri deprem şiddeti ve deprem konumu süzgecine göre Telegram isimli mesajlaşma uygulaması ile haberdar olmasını sağlamaktır. Uygulamada veri kaynağı olarak Boğaziçi Üniversitesi Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü’nün yayınladığı güncel deprem verileri kullanılmıştır. Çalışmada Telegram uygulaması, sağladığı uygulama geliştirme araçları ve sohbet botu yetenekleri nedeniyle seçilmiştir.



Şekil 7: Sunucusuz mekânsal veri analizi servisi tasarımı ve bileşenleri

Depremler veri kaynağından ‘Deprem Verisi Okuyucu’ isimli sunucusuz çalışan fonksiyon ile belli zaman aralıkları ile okunmaktadır. Eğer en son okunma zamanından sonra yeni bir deprem olmuşsa bu depremden haberdar olacak abonelerin tespiti için sunucusuz yapısal veri depolama hizmeti olan Amazon DynamoDB servisinden tüm aboneler sorgulanır. Yeni deprem şiddeti ve konumu ile abonelerin belirlediği deprem şiddeti ve konumu süzgeçleri karşılaştırılır. Karşılaştırma sonucunda haberdar edilecek her bir abone için bildirim emri ayrı ayrı mesaj gönderim kuyruğuna yazılmaktadır. Mesaj kuyruğu AWS firmasının sunduğu Amazon SQS adında sunucusuz bir uygulama servisi. ‘Bildirim Gönderici’ isimli fonksiyon HTTP protokolü üzerinden Telegram uygulamasının internet servislerini kullanarak abonelere bildirim mesajını gönderir (Şekil 8). Bildirim aboneleri aynı zamanda bildirimleri ileten Telegram sohbet botunun da kullanıcılarıdır. Bu bot ile hem bildirimleri almakta hem de süzgeç ayarlarını yapabilmektedirler. Bot üzerinden yapılan yeni abonelik, abonelik silme ve süzgeç güncelleme işlemleri ‘Telegram Bot’ isimli fonksiyon yardımıyla yapılır. Kullanıcıların gönderdiği deprem

konumu süzgeci coğrafi nokta verisi olarak ‘Telegram Bot’ isimli fonksiyona iletilir ve Amazon DynamoDB servisinde GeoHash kodlaması ile kaydedilir.



(a)

(b)

Şekil 8: Telegram botundan gelen uyarılar (a) ve kullanıcı etkileşimi (b)

Tasarım, açık kaynak kodlu olarak geliştirilmiş ve sosyal kod paylaşım platformu olan GitHub internet sitesine yüklenmiştir (URL-14). Sistem AWS bulut bilişim sağlayıcısı üzerinde 29 Ekim 2019 tarihinden bu yana çalışmaktadır. Sistemin bu süre boyunca hiç bakım gerekmeden çalıştığı gözlenmiştir (Şekil 9). Yapılan kod iyileştirmeleri dışında alt yapının yönetilmesi gerekmemiştir. Güncel kullanıcı sayısı 3200’dür. AWS bulut bilişim sağlayıcısı bir fonksiyonun ilk bir milyon çalışmasından sonra ücretlendirmeye başlamaktadır. Sistem bugüne kadar bu kullanım miktarına erişmediği için bir maliyeti olmamıştır.

Function name	Description	Package type	Runtime	Code size	Last modified
KEarthquakePuller	Earthquake puller and distributor. Pulls data from Kandilli website.	Zip	.NET Core 2.1 (C#/PowerShell)	1.9 MB	1 year ago
KEarthquakeBot	Earthquake bot to notify subscribers based on their preferences.	Zip	.NET Core 2.1 (C#/PowerShell)	1.1 MB	1 year ago
KEarthquakeNotifier	Earthquake notifier to Telegram subscribers	Zip	.NET Core 2.1 (C#/PowerShell)	1.1 MB	1 year ago

Şekil 9: Deprem habercisi sunucusuz mimarideki uygulamasının bir sene boyunca hiç değiştirilmeden çalıştığını gösteren AWS Lambda portalı ekran görüntüsü

5. Sonuç ve Öneriler

Önümüzdeki yıllarda sunucusuz teknolojilerle geliştirilen uygulamaları bulut bilişim sağlayıcıları üzerinde ve yerel bilişim teknolojileri altyapılarında daha fazla göreceğiz. Bu çalışmada coğrafi bilgi teknolojilerinin veri toplama, veri analizi ve verinin sorgulaması konularında sunucusuz teknolojilerle birlikte kullanıldığı sistemlerin tasarım önerileri verilerek incelenmiştir. Bulut bilişim pazarındaki diğer oyuncuların sunduğu hizmetlerin birbirlerine çok benzer oldukları görülmüştür (Jimenez & Zeinzer, 2018). Bu yüzden bu çalışmadaki sistem tasarımları her ne kadar sadece bir bulut bilişim sağlayıcı firmanın sunduğu sunucusuz hizmetlerle tasarlanmış olsa da diğer bulut bilişim sağlayıcıları ile uyumlu olarak çalışacaktır. Bununla birlikte bulut bilişim servis sağlayıcısı kullanılmak istenmediği durumda bir sunucusuz hesaplama ve depolama

altyapısı, yerel bilişim teknoloji altyapısı üzerinde çeşitli konteyner yönetim araçları ile kurgulanabilir. Böylece verinin bulut bilişim sağlayıcısı üzerine taşınmasının mümkün olmadığı durumlarda da bu teknolojiden yararlanılabilir.

FaaS servisleri kullanıcılarından sadece kod parçasını yüklemesini beklediği için; bir programlama dili konusunda yetkin bir coğrafi bilgi teknolojileri uzmanı, ileri seviyede bilişim teknolojileri altyapısı yönetim konularına hakim olması gerekmeden hazırladığı algoritmayı sunucusuz mimarilerle diğer bulut bilişim modellerine göre daha kolay çalıştırabilir. İstenirse bu algoritmanın diğer bulut bilişim servisleriyle ortak bir iş akışı ile çalışması da sağlanabilir. Ancak yine de fonksiyonların diğer servislerle olan iletişimi çalışma süresini arttırabileceği için sistem tasarımında birlikte çalışabilirlik konusunun etkileri göz ardı edilmemelidir. Harcanan fazladan her milisaniyenin sunucusuz bulut bilişim servislerinin ücretlendirilmesinde olumsuz yönde katkısı olduğu görülmüştür. FaaS servislerinin çalışma süresi ve sunulan donanım kapasitesi limitleri nedeniyle mekânsal zekâ uygulamaları için CaaS servislerinin daha uygun olduğu görülmüştür.

Hâlihazırdaki çalışan bir mekânsal bilişim sistemini sunucusuz mimariye geçirirken kullanılacak teknolojilerin avantajları ve dezavantajları iyi değerlendirilmelidir. Örneğin makalede de bahsedilen soğuk başlama süresi uygulamanın ilk cevap verme süresini uzatacağından kullanıcı deneyimini olumsuz etkileyebilir. Diğer bir örnek ise yerel ağda sunucusuz mimariyle ilişkisel veri saklamak için hazır bir çözüm bulunmamaktadır. Ancak bu türden bir çözümün konteyner orkestrasyon yazılımları ile elde edilebileceği görülmüştür.

Çalışma kapsamında verilen deprem habercisi örneği gelecek çalışmalarda örnek olarak kullanılabilmesi için açık kaynak kodlu olarak yayınlanmıştır. Bu örnek aynı zamanda birden fazla FaaS'in birlikte bir iş akışı içerisinde nasıl çalışabildiğini de göstermiştir. Bu sayede mekânsal veri işleme iş akışları sunucusuz hesaplama ve veri depolama teknolojileri kullanılarak geliştirilebilir.

Sonuç olarak bu çalışmada coğrafi bilgi teknolojileri konusunda araştırma yapanların ve kullanıcılarının sunucusuz bilişim teknolojileri konusunda farkındalıklarının arttırılması hedeflenerek bu teknolojinin coğrafi bilgi teknolojileri ile ilişkisi incelenmiştir.

Çıkar Çatışması Beyanı

Yazarlar, bu çalışmada bilinen ilgili herhangi bir finansal veya finansal olmayan çıkar çatışması olmadığını beyan eder.

Yazar Katkısı

Mete Ercan Pakdil: Fikir, Tasarım, Literatür taraması, Yazım. **Rahmi Nurhan Çelik:** Denetleme, Analiz ve yorumlama, Makale değerlendirme.

Kaynaklar

- Adzic, G., & Chatley, R. (2017). Serverless computing: economic and architectural impact. *Proceedings of the 2017 11th joint meeting on foundations of software engineering*, 884-889.
- Amirian, P., Basiri, A., & Winstanley, A. (2014). Evaluation of data management systems for geospatial big data. *International Conference on Computational Science and Its Applications*, 678-690.
- Anand, S., Johnson, A., Mathikshara, P., & Karthik, R. (2019). Real-time GPS tracking using serverless architecture and ARM processor. *2019 11th International Conference on Communication Systems & Networks*, 541-543.

- Audevart, A., Banachewicz K., & Massaron, L. (2021). *Machine Learning Using TensorFlow Cookbook*. Packt Publishing.
- Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., & Suter, P. (2017). Serverless computing: Current trends and open problems. *Research advances in cloud computing*, 1-20.
- Barbieri, L., & Bonanni, M. (2019). *Mastering Azure Serverless Computing*, Packt Publishing.
- Bateria, C. (Ed.) (2012). *Cartography - A Tool for Spatial Analysis*, InTech Open.
- Bebortta, S., Das, S. K., Kandpal, M., Barik, R. K., & Dubey, H. (2020). Geospatial serverless computing: Architectures, tools and future directions. *ISPRS International Journal of Geo-Information*, 9(5), 311.
- Chapin, J., & Roberts, M. (2020). *Programming AWS Lambda*, O'Reilly Media Inc.
- Chowhan, K. (2018). *Hands-On Serverless Computing*, Packt Publishing.
- Dold, J., & Groopman, J. (2017). The future of geospatial intelligence. *Geo-spatial information science*, 20(2), 151-162.
- Evans, M. R., Oliver, D., Yang, K., Zhou, X., Ali, R. Y., & Shekhar, S. (2019). Enabling spatial big data via CyberGIS: Challenges and opportunities. *CyberGIS for geospatial discovery and innovation*, 143-170.
- Jimenez, V. E., & Zeiner, H. (2018). Serverless Cloud Computing: a Comparison between "Function as a Service" Platforms. *7th International Conference on Information Technology Convergence and Services*, Vol. 5, 15-22.
- Flasher, J. (2019). *Best practices for working with large-scale geospatial data*. AWS reinvent.
- Gratier, T., Spencer, P., & Hazzard, E. (2015). *OpenLayers 3: Beginner's Guide*: Packt Publishing.
- Güney, C., & Çelik, R.N. (2020). Harita ve kadastro mühendisliğinin dijital ekosistemde hayatta kalabilmesi için paradigma değişimi: mekânsal zekâ. *17. Harita Bilimsel ve Teknik Kurultayı*, 2020.
- Luksa, M. (2017). *Kubernetes in action*. Manning Publications.
- Mete, M. O., & Yomralioğlu, T. (2021a). Implementation of serverless cloud GIS platform for land valuation. *International Journal of Digital Earth*, 14(7), 836-850.
- Mete, M. O., & Yomralioğlu, T. (2021b). Açık Kaynaklı Bulut CBS Yardımıyla Kitlemel Taşınmaz Değerleme Uygulaması. *Harita Dergisi*, 165, 28-42.
- Nickoloff, J., & Kuenzli, S. (2019). *Docker in action*. Manning Publications.
- Pakdil, M. E., & Çelik, R. N. (2021). Serverless Geospatial Data Processing Workflow System Design. *ISPRS International Journal of Geo-Information*, 11(1), 20.
- Patterson, S. (2019). *Learn AWS Serverless Computing: A Beginner's Guide to Using AWS Lambda, Amazon API Gateway, and Services from Amazon Web Services*. Packt Publishing.
- Siddiqui, T., Siddiqui, S. A., & Khan, N. A. (2019). Comprehensive analysis of container technology. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 218-223.
- Tang, W., & Wang, S. (Eds.). (2020). *High performance computing for geospatial applications*. Springer International Publishing.
- van Eyk, E., Toader, L., Talluri, S., Versluis, L., Uță, A., & Iosup, A. (2018). Serverless is more: From paas to present cloud computing. *IEEE Internet Computing*, 22(5), 8-17.
- Vohra, D. (2018). *Amazon Fargate Quick Start Guide: Learn how to use AWS Fargate to run containers with ease*. Packt Publishing.
- Wan, L., Huang, Z., & Peng, X. (2016). An effective NoSQL-based vector map tile management approach. *ISPRS International Journal of Geo-Information*, 5(11), 215.
- Zaragozí, B. M., Trilles, S., & Navarro-Carrión, J. T. (2020). Leveraging container technologies in a GIScience project: a perspective from open reproducible research. *ISPRS International Journal of Geo-Information*, 9(3), 138.
- URL-1: Schachar, A. (2019). *Serverless CaaS: Rethinking app infrastructure*. <https://spot.io/blog/serverless-caas-rethinking-app-infrastructure/>, (Erişim Tarihi:27 Mayıs 2021).
- URL-2: CNFC Serverless Working Group Whitepaper, https://github.com/cnfc/wg-serverless/blob/master/whitepapers/serverless-overview/cnfc_serverless_whitepaper_v1.0.pdf, (Erişim tarihi: 3 Şubat 2022).
- URL-3: Ruparathna, R. (2019), *What Makes a Storage Service Truly Serverless?*, <https://dzone.com/articles/what-makes-a-storage-service-truly-serverless> , (Erişim Tarihi: 27 Mayıs 2021).
- URL-4: Moten D. (2019), *Turn Amazon S3 into a spatio-temporal database*, <https://towardsdatascience.com/turn-amazon-s3-into-a-spatio-temporal-database-40f1a210e943>, (Erişim Tarihi: 23 Mayıs 2021).

- URL-5: Buchanan, I., *What is Containers as a Service?*, <https://www.atlassian.com/continuous-delivery/microservices/containers-as-a-service>, (Erişim tarihi: 3 Şubat 2022).
- URL-6: MapServer Open Source Platform, <https://hub.docker.com/t/mapserver/mapserver>, (Erişim Tarihi: 3 Şubat 2022).
- URL-7: Docker, <https://www.docker.com>, (Erişim tarihi: 3 Şubat 2022).
- URL-8: Beswick J. (2020), *Implementing geohashing at scale in serverless web applications*, <https://aws.amazon.com/blogs/compute/implementing-geohashing-at-scale-in-serverless-web-applications/>, (Erişim Tarihi: 23 Mayıs 2021).
- URL-9: Mapbox Vector Tile Specification, <https://github.com/mapbox/vector-tile-spec>, (Erişim tarihi: 3 Şubat 2022).
- URL-10:Tegola, <https://tegola.io/>, (Erişim tarihi: 3 Şubat 2022).
- URL-11: Cloud optimized GeoTIFF in depth, <https://www.cogeo.org/in-depth.html>, (Erişim tarihi: 23 Mayıs 2021).
- URL-12: MapProxy, <https://mapproxy.org/>, (Erişim tarihi: 03 Şubat 2022).URL-13: MapProxy, <https://mapproxy.org/>, (Erişim tarihi: 3 Şubat 2022).
- URL-13: Ivanovic B. & Ivanovic Z. (2017), *How to deploy deep learning models with AWS lambda and TensorFlow*, <https://aws.amazon.com/blogs/machine-learning/how-to-deploy-deep-learning-models-with-aws-lambda-and-tensorflow/>, (Erişim Tarihi: 23 Mayıs 2021).
- URL-14: Kandilli Deprem Habercisi, <https://github.com/Geomates/KandilliEarthquakeNotifier>, (Erişim tarihi: 4 Şubat 2022).