**T.C.**

**SOSYAL GÜVENLİK KURUMU BAŞKANLIĞI**

**MİKROSERVİS MİMARİSİNİN İNCELENMESİ: BİR SOSYAL GÜVENLİK KURUMU YAZILIM UYGULAMASI ÖRNEĞİ**

**Sosyal Güvenlik Uzmanlığı Tezi**

**Hazırlayan**

**Hallaç Mansur AKBAŞ**

**Tez Danışmanı**

**Murat BİNAY**

**Sosyal Güvenlik Uzmanı**

**Ağustos 2024**

**Ankara**

Hallaç Mansur AKBAŞ tarafından hazırlanan bu çalışmanın Sosyal Güvenlik Uzmanlığı Tezi olarak uygun olduğunu onaylarım. …./…./2024

Tez Danışmanının Adı SOYADI: Murat BİNAY

İmzası:

Bu çalışma, Kurulumuz tarafından oy birliği / oy çokluğu ile Sosyal Güvenlik Uzmanlığı Tezi olarak kabul edilmiştir.

|  |  |  |
| --- | --- | --- |
|  | **Adı Soyadı ve Unvanı** | **İmzası** |
| Başkan | : ………………………………………………… | …....………... |
| Üye | : …………………………………………………. | …....………... |
| Üye | : ………………………………………………… | …....………... |
| Üye | : …………………………………………………. | …....………... |
| Üye | : ………………………………………………… | …....………... |

Tarih: …./…./………

# 

# TEŞEKKÜR

Bu çalışma sırasında tecrübe ve bilgisi ile beni yönlendiren tez danışmanım Emre MUTLU’ya, ihtiyaç duyduğum konularda desteklerini esirgemeyen çalışma arkadaşlarım Hasan İbrahim KOÇAK’a ve Ramazan SARIALTIN’a, ayrıca her koşulda yanımda olan anneme, babama ve kardeşlerime teşekkürü borç bilirim.

**Ankara 2024 Hallaç Mansur AKBAŞ**

# BEYAN

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde akademik kurallara ve bilimsel etik davranış ilkelerine uyduğumu, ayrıca, bu kural ve ilkelerin gereği olarak, çalışmada bana ait olmayan tüm veri, düşünce ve sonuçları andığımı ve kaynağını gösterdiğimi beyan ederim.

…./…./……..

Adı ve SOYADI: Hallaç Mansur AKBAŞ

İmzası:

# ÖZET

**Sosyal Güvenlik Uzmanlığı Tezi**

**MİKROSERVİS MİMARİSİNİN İNCELENMESİ:**

**BİR SOSYAL GÜVENLİK KURUMU YAZILIM UYGULAMASI ÖRNEĞİ**

**Hallaç Mansur AKBAŞ**

İnce istemci, uzak birimlerdeki kullanıcı ihtiyaçlarını zamanla izler ve uzak birime merkezdeki kaynaklardan gerekli ve yeterli miktardakini ayırarak etkin çözümlerle ve daha az kaynak kullanımıyla önemli bir iyileştirme sağlar.

İnce istemci sanallaştırma teknolojisi kullanır. Sanallaştırma; fiziksel olarak merkezde olan büyük bir havuzu (yazılım sayesinde) ihtiyaç olduğu kadar mantıksal birimlere bölerek kaynakları daha verimli hale getirmektir. Böylece eskiden her birimde ayrı ayrı bulunan fiziksel kaynaktaki kullanılmayan kapasite daha etkin bir biçimde kullanılacak ve merkezi yönetimle sistem performansı kat kat artacak ve yönetim maliyetinin de düşük olmasını sağlayacaktır.

SGK, sunucu sanallaştırma altyapısı ile bütün hizmetlerini kamuya sunmaktadır. Sunucu sanallaştırma, merkezdeki etkinliği sağlamakla birlikte mevcut taşra altyapılarında bir sanallaştırma altyapısı olmadığından buralardaki performanslara doğrudan bir etki yapamaz. İşte tam da burada ince istemci mimarisi ve alt yapısının gerekliliği ortaya çıkmaktadır. Bu çalışmada ilk olarak mevcut sanallaştırma altyapısı incelenmiş, topolojisi çıkartılmış ve bileşenleri anlatılmıştır. Ardından ince istemci mimarisinin SGK taşra birimleri kullanıcı masa üstü ortamına uygulanabilirliği test edilmiştir ve olumlu sonuçlar alınmıştır. Ayrıca Bilişim Ağı, Enerji Verimliliği, Yönetim Politikası ve Felaket Kurtarma konularında kurum bilişim teknolojileri altyapısında uygulanabilecek çözümler belirlenmiştir. Tezin amacı İnce İstemci tekonolojileri kullanımı halinde Kurum’da haberleşme yönetim ve yedekleme açısından ortaya çıkan faydaların değerlendirilmesi ve ince istemci ile ilgili bilinç oluşmasıdır.

**Anahtar Kelimeler:** Mikroservis, Yazılım Mimarisi, Monolitik Mimari, Yazılım Tasarımı, Ölçeklenebilirlik, Konteynerizasyon, DevOps

# ABSTRACT

**Social Security Expertise Thesis**

**EXAMINATION OF MICROSERVICE ARCHITECTURE: A SAMPLE OF A SOCIAL SECURITY INSTITUTION SOFTWARE APPLICATION**

**Hallaç Mansur AKBAŞ**

Thin client, the remote unit monitor user needs over time and with the necessary and adequate amount of resources in the next one effective solution by separating the central resources and remote unit it provides a significant improvement with less resource use.

Thin client uses virtualization technology. Virtualization, with a large pool in the center physically (thanks to software) to logically divide up our resources that we need to become more efficient units. Thus enabling each unit formerly unused capacity on the physical resources available separately to be used more effectively and will increase central management with system performance and management many times that of the low.

SGK, offers all the services to the public with the server virtualization infrastructure. Server virtualization; while providing a direct impact on activity in the center performance can’t be a direct impact on province performance as there is not a virtualization infrastructure around the existing provincial infrastructure. It is arises here is exactly the thin client architecture and requirements of the infrastructure. In this study first of all, examined the existing virtualization infrastructure, topology designated and components described. Then SGK provincial units of the thin client architecture have been tested the applicability of the user desktop environment and positive results were. In addition, IT network, energy efficiency, management policies and disaster recovery solutions identified are applicable to corporate information technology infrastructure. The purpose of this thesis is if the use of thin client technologies at the institution, evaluation of the benefits in terms of communication management and backup and aimed to create awareness about thin client.

**Key Words:** Microservice, Software Architecture, Monolithic Architecture, Software Design, Scalability, Containerization, DevOps

# YÖNTEM

Bu çalışmada ulusal ve yabancı kaynaklardan faydalanılarak hazırlanmıştır. Sanallaştırmanın ve sanallaştırma türlerinin detaylı araştırılmasında yurtiçi ve yurtdışındaki makalelerden, analizlerden, dergi ve kitaplardan, kurumların yapmış oldukları çalışmalardan yararlanılmıştır. Ayrıca konu ile ilgili ulusal ve yabancı internet sitelerinin de ağırlıklı olarak kullanımı düşünülmektedir. Tezin yazımının üç aşamadan meydana gelmesi düşünülmektedir.

Birinci bölümde Sosyal Güvenlik Kurumu bilgi işlem ağı ve altyapısı, ince istemcinin temeli olan sanallaştırma kavramanın tarihçesinden, kurum veri merkezlerinden ve hali hazırda yürütüyor olunan projelerden bahsedilecektir. Ayrıca diğer sanallaştırma yöntemleri olan; sunucu sanallaştırma, masaüstü sanallaştırma, uygulama sanallaştırma, veri depolama sanallaştırma, ağ sanallaştırma, veri ve veritabanı sanallaştırma, bellek sanallaştırma gibi sanallaştırma türleri detaylı bir şekilde incelenecektir.

İkinci bölümde ise ince istemcinin ne olduğu, mimarisi, uygulama alanları ve ince istemci ile sanallaştırma modeli uygulanırken, hangi hususlara dikkat edilmesi gerektiği açıklanmıştır. Ayrıca ince istemcinin geleneksel masaüstü bilgisayarlardan mantıksal farkı ve ince istemciye geçiş süreci de yine bu bölümde incelenecektir.

Son bölümde ise ince istemci teknolojisi kullanılarak taşralarda sanallaştırmaya geçiş sonucu elde edilebilecek faydalar ve geçiş süreci anlatılmıştır. Yapılması düşünülen yeni sanallaştırma teknolojisinin Sosyal Güvenlik Kurumu’na olası getirileri riskleri ve götürüleri üzerinde durulmuştur. Ayrıca ince istemci teknolojisi kullanılarak oluşturulan sanallaştırmanın ne gibi güvenlik getirileri olduğu ve enerji verimliliği açısından (otuz bin kullanıcılı üç yüz ayrı ve uzak lokasyonda bulunan bir kurumda) ne gibi getirileri olduğu da bu bölümde anlatılmıştır. Ayrıca pilot bölge seçilen bir taşra biriminde birkaç kullanıcının ince istemci kullanarak günlük işlerini yapması olanağını oluşturup, bu durumun sonuçlarını kurum geneline göre detaylı analiz edilmesi düşünülmekte ve ince istemci teknolojisinin kullanıldığı sanal bir platform kurup bu platform üzerinden günlük işler yapıp buna göre bir inceleme yapılması düşünülmektedir.

**İÇİNDEKİLER**

**ŞEKİLLER LİSTESİ**

**TABLOLAR LİSTESİ**

**KISALTMALAR**

**GİRİŞ**

Günümüzün hızla gelişen ve değişen teknoloji ortamında yazılım geliştirme süreç ve yöntemleri ihtiyaçlara cevap verebilmek için sürekli bir dönüşüm içindedir. Hız, performans ve depolama kapasiteleri artan bilgisayar sistemleri daha karmaşık ve güçlü yazılımların geliştirilmesine imkan sağlamıştır. Bunun sonucunda üretilen büyük ölçekli yazılımları geliştirmek ve yönetmek için yeni mimari yaklaşımlar ortaya çıkmıştır.

Klasik yazılım geliştirme mimarisinde tek bir kod tabanı üzerinde geliştirme yapılır ve geliştirilen yazılım bir bütün halinde kullanıma alınır. Özellikle büyük ölçekli yazılımların geliştirilmesi ve yönetilmesi konusunda bu yaklaşım bazı sınırlamaları ve zorlukları beraberinde getirir. Bu mimarinin getirdiği zorlukları aşmak ve daha esnek, ölçeklenebilir ve yönetilebilir yazılım çözümleri sunabilmek için farklı bir mimari model olan mikroservis mimarisi ortaya çıkmıştır.

Mikroservis mimarisinin kullanımı Netflix, Amazon, Spotify gibi dünyadaki önemli kurumsal şirketlerin bu mimariyi kullanmada öncülük etmesiyle yazılım dünyasında yaygınlaşmıştır. Bu mimari büyük monolitik uygulamaların yerine küçük, bağımsız ve birbiri ile entegre çalışabilen servislerden oluşan bir uygulama yapısı önermektedir. Her bir mikroservis, belirli bir işlevi gerçekleştirir ve bu servisler bir araya gelerek büyük uygulamaları oluşturur.

SGK bünyesinde hizmet veren yazılımlar, milyonlarca insanın hayatını doğrudan etkileyen kritik öneme sahip yazılımlardır. Bu sistemlerin doğru bir şekilde tasarlanması ve geliştirilmesi hem kurumun işleyişi hem de hizmet verilen vatandaşlar için büyük önem taşımaktadır. Gerek büyük uygulama yazılımlarının karmaşıklığının üstesinden gelmek gerekse yazılım ekiplerinin rahat ve uyumlu bir şekilde çalışabilmesini sağlamak için mikroservis mimarisi değerlendirilecektir. Avantajları ve dezavantajları göz önüne alınarak kurumun kullanımına uygunluğu hem teknik açıdan hem kültürel açıdan ele alınacaktır.

**YEDİNCİ BÖLÜM**

**UYGULAMA ÖRNEĞİ**

X dağıtık sistemlerde veri bütünlüğünü sağlamak önemli. Bu yüzden 2PC ve sga yöntemi bunu sağlamaya yöneliktir. xxxxxxxx xxxxxxxxxxxxx xxxxxxx xxxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

1. **KURUM BİLGİ İŞLEM MİMARİSİ**
   1. **Aaaaaa aaaaaaaaa aaaaaaaaa**

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

* 1. **Bbbbbbbbb bbbbbb**

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**1.2.1. Ccccccc cccccc cccccccc ccccccccccccccccccc**

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

**1.2.2. Ddddddddddd**

**1.2.2.1. Ddddddddddd**

# SONUÇ ve ÖNERİLER

4.3 Case Studies

4.3.1 Review of Similar Transformations: Summarize case studies where similar institutions have adopted microservice architectures.

4.3.2 Lessons Learned: Extract and discuss key learnings that could be relevant to a social security institution.

6. Expert Insights

6.1 Interviews with Industry Experts

6.1.1 Insights on Microservices in Public Sector: Synthesize opinions and experiences from professionals who have overseen similar transitions.

6.2 Panel Discussion Summaries

6.2.1 Recorded Discussions: Summaries of discussions with a panel of IT architects and system developers from the public sector.

* 1. **Bbbbbbbbb bbbbbb**

Rest API, Feign rest çağrısı yapmakta iyi ayrıca load balancing içinde yardımcı oluyor, Load balancing için Eureka da işin içine giriyor. Instance’ler arttıkça 8000 8001 8002 yük dengelemesi yapabilmek isteriz. Bunun için mikroservis mimarisindeki tüm mikroservislerin tüm instanceleri servis keşfi’nde tutulur. Test prod ortamları