T.C.
OKAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KULLANICI KİŞİSEL VERİLERİNDE PLATFORM BAĞIMSIZ GÜVENLİK

Zana İLHAN

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
Yard. Doç. Dr. Bahri Atay ÖZGÖVDE

İSTANBUL, Mayıs 2012

Bu çalışma 26/01/2012 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Danışman Adı :Yard.Doç.Dr. Bahri Atay ÖZGÖVDE

Üniversitesi : Galatasaray Üniversitesi

Fakültesi : Mühendislik ve Teknoloji Fakültesi

Jüri Adı : Prof.Dr.B.Tevfik AKGÜN

Üniversitesi : Okan Üniversitesi

Fakültesi : Mühendislik Mimarlık Fakültesi

Jüri Adı : Yard.Doç.Dr. Fatih ÖZAYDIN

Üniversite : Okan Üniversitesi

Fakülte : Mühendislik Mimarlık Fakültesi

ÖNSÖZ

Yüksek lisans öğrenimim ve tez çalışmalarım boyunca gösterdikleri her türlü destek ve yardımdan dolayı çok değerli hocalarımYard. Doc. Dr. Bahri Atay ÖZGÖVDE ve Prof.Dr.Tevfik AKGÜN'e en içten dileklerimle teşekkür ederim. Ayrıca bu zorlu süreçte hem teknik hemde manevi desteklerini benden esirgemiyen ARDIC A.Ş. ye ve sevgili eşim Fatoş'a da yardımlarından dolayı ayrıca teşekkürü bir borc bilirim.

İSTANBUL, Mayıs 2012 Zana İLHAN

İÇİNDEKİLER

ŞEKİL LİSTESİ

TABLO LİSTESİ

KISALTMA LİSTESİ

PIPSU : Personal Security For User Private Data
SDK : Software Development Kit
IPC : Inter Process Communication
SQL : Yapısal Sorgulama Dili(Structured Query Language)
ICC : Internal Component Communication
RSS : Really Simple Syndication
GPS : Global Positioning System
GUI : Graphical User Interface
PIM : Personel Information Management

ÖZET

KULLANICI KİŞİSEL VERİLERİNDE PLATFORM BAĞIMSIZ GÜVENLİK

Günümüzde mobil aygıtlar, hayatımızın bir çok alanında aktif olarak kullanılmaya başlanmıştır. Taşınabilir formda ve gündelik hayata tümleşik biçimde kullanılan bu aygıtlar artan hesaplama kapasiteleri ile kişisel bilgisayarların yerini almaya başlamışlardır. Bu bağlamda kullanıcılarıngün geçtikçe daha önemli bilgilerini barındıran bu aygıtlar için güvenlik önemli bir sorundur. Yazılım ve donanım mimarilerinde tam olarak standartlaşmaya gidilememiş olması, uygulamaların yüksek yetkilerle sisteme yüklenmesi, uygulama dükkanları tipi yazılım üretim ve tüketim süreçlerinin henüz oturmamış olması gibi etmenler yüzünden mobil platformlarda güvenlik sorunlarının artarak yaşanacağını öngörülmektedir.

Bu çalışmada mobil platformlarda karşılaşılan güvenlik problemlerinin sebepleri incelenmiş, boyutları ortaya konulmuş ve bir çözüm önerisi getirilmiştir. Örnek platform olarak gittikçe artan pazar payı ile baskın hale gelmeye başlayan Android işletim sistemi ele alınmış ve Android mimari yapılanması içinde tüm uygulama türleri için geçerli olabilecek bir çözüm önerisi olarak PIPSU(Personel Security For User Private Data) ortaya konulmuştur.PIPSU ile mobil platform kullanıcılarının kişisel bilgilerinin, kullanıcı kontrolü olmaksızın kullanılmasının önüne geçilmesini hedeflenmiştir.

PIPSU, kullanıcı kişisel bilgilerinin iki katmanda güvenli şekilde saklamayı hedefleyen bir çözüm önerisidir. Önerilen güvenlik yaklaşımı mimari olarak mobil ortam istemcisi (PIPSU-L) ve bulut tabanlı sunucu (PIPSU-C) biçiminde yapılandırılmıştır. PIPSU-L ile fiziksel mobil aygıt üzerindeki kişisel bilgilerin kullanımı, kontrolü ve yapılandırılması, PIPSU-C ile de, kişisel bilgilerin mobil ortam bağımsız olarak güvenli şekilde saklanması hedeflenmiştir. PIPSU-L sahip olduğu Launcher yapısı ile mobil ortam kullanıcılarının kullanım kiplerini değiştirmeden, kişisel verileri önce aygıt üzerinde geliştirilen depolama çözümüne, daha sonrada bu verileri farklı mobil ortamlara taşıyabilmek amacıyla PIPSU-C bileşenine yönlendirmektedir.

Bu çalışma kapsamında yapılan analizler sonucu ortaya çıkan çözüm modeli ile kullanıcı kişisel bilgilerinin güvenli şekilde mobil platformda kullanılabilmesi hedeflenmiştir. PIPSU'nun en önemli artılarından biri, getirdiği yüksek güvenlik seviyesinin kullanıcıların mobil aygıtları ile gündelik kullanım alışkanlıklarını ve etkileşim kiplerini değiştirmek zorunda kalmadan etkili kılmasıdır. Bu durum getirilen önerinin önemli bir güvenlik çözümü olmasının dışında pratik ve ticari uygulanabilirliği açısından önem arz etmektedir.

SUMMARY

PLATFORM INDEPENDENTSECURITY FOR USER PRIVATE DATA

Mobile devices nowadays are signaling the start of a new era in personal computing. These devices, with their increasing computational capabilities, easy-to-carry form factors and always-on network connections are heavily integrated into our daily lives.It can easily be forecasted that in near future they will replace traditional personal computing devices such as PCs and regular laptops. In that respect, the characteristics of the data these devices carry is evolving from simple contact lists to all forms of critical and/or confidential personal data.

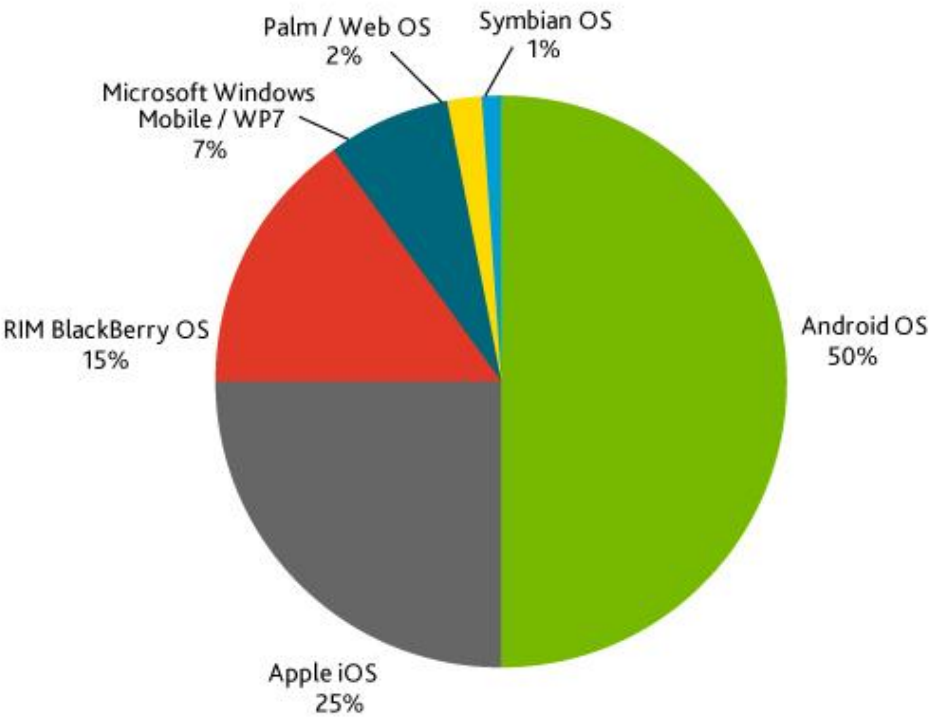
However, this increasing responsibility assigned to mobile devices in terms of keeping personal data is not supported by enough security measures and serious threats exist for the casual users. Hardware and software architectures still being in the process of standardization, applications being installed with high authorization, low quality and non-monitored software development environment led by the app stores are among the factors that contribute to the security problems faced by the users.

In this thesis, security threats to user private data on mobile platforms examined in all relevant aspects and a solution architecture is proposed. Among many mobile platforms currently exist today, with its dominating and increasing market share Android mobile OS is chosen as the implementation platform for our proposed solution PIPSU.PIPSU aims to prevent any access to private data without user approval.

PIPSU architecture is based on a mobile client (PIPSU-L) and a cloud based server (PIPSU-C). Due to the way PIPSU operates, it provides a high level protection for the user private data without affecting the interactivity mode of the user with the mobile device. This fact makes an important distinction between PIPSU and similar mobile data security solutions in terms of practical usability.

1. GİRİŞ

Günümüzde mobil platformlar bir çok ihtiyacımızı karşılayabilen, zamanımızın önemli bir kısmını geçirdiğimiz birincil hesaplama aygıtları haline gelmişlerdir. Akıllı telefonlar, tabletler, eePCler, laptoplar ve benzer mobil aygıtlar yarattıkları kolaylıklar ve yenilikler sayesinde, büyük bir kullanıcı kitlesi tarafından kullanılmaktadırlar ve hayatımızda vazgeçilmez bir hal almışlardır. Donanım ve yazılım destekleri ile gelen yeni mobil uygulamalar sayesinde mobil aygıtlar eğlence, iletişim ve ticari alanlarda yeni kullanım pazarları açmışlardır. Mobil platformların pazar payları ve kullanıcı kitlelerine dair Nielsen'in ağustos 2011 için yayınladığı rapor Şekil 1.1'de özetlenmişir[1].



Şekil 1.1Ağustos 2011 Android Pazar Payları

Bu rekabette sadece işletim sistemlerinin ve/veya cep telefonu donanımlarının özellikleri değil platformların sunduğu uygulamaların çeşitliliği de önemli rol oynamaktadır. Piyasadaki mobil platformlar arasında Android işletim sisteminin çok kısa zamanda pazarda ulaştığı nokta dikkat çekicidir ve yakın gelecekte pazarda daha büyük yere sahip olacağı öngörülmektedir.Android ve mobil aygıtlar pazarının diğer büyük oyuncularının sunduğu uygulama marketleri ve bu marketlerdeki uygulamaların çeşitliliği ve zenginliği de bu açıdan platform rekabetinde etkili olmaktadır. Ekim 2011 itibariyle itibariyle uygulama marketlerinin başta gelenlerinden Apple App Storeda 472,937 civarinda uygulama bulunmaktadır, benzer şekilde Android Market ise 600.000 den fazla uygulama ve pazarın önemli oyuncularından BlackBerry ise App World ile 10.000 uygulamaya sahiptir.Bütün uygulama marketleri içinde uygulama sayısı en hızla artan Android uygulamaları olup, Eylül 2011 itibariyle bu konuda birinci olacağı öngörülmektedir[2].

Tablo 1.1 Gartner Firmasının Haziran 2011 Market Raporu[26,27,28]

Dünya çapında mobil işletim sistmelerin Gartner araştırma firmasının 2009-2015 yıllarına göre tahmini market payları					Mobil işletim sistemlerinin IDC araştırma şirketinin 2011 -2015 yılları için yaptığı çalışmaya göre tahmini büyüme oranları		
İşletim Sistemleri	2009 market payı	2010 market payı	2011 market payı	2015 market payı	2011 market payı	2015 market payı	2011-2015 CAGR
Android	3.9%	22.7%	38.5%	48.8%	39.5%	45.4%	23.8%
BlackBerry	19.9%	16.0%	13.4%	11.1%	14.9%	13.7%	17.1%
iOS	14.4%	15.7%	19.4%	17.2%	15.7%	15.3%	18.8%
Symbian	46.9%	37.6%	19.2%	0.1%	20.9%	0.2%	-65.0%
WindowsMobile	8.7%	4.2%	5.6%	19.5%	5.5%	20.9%	67.1%
Diğerleri	6.1%	3.8%	3.9%	3.3%	3.5%	4.6%	28.0%
Toplam Satışlar	172 milyon	297 milyon	468 milyon	631 milyon	450 milyon		19.6%

Mobil platformlar kısa sürede büyük kullanıcı kitlelerinin vazgeçilmezi haline gelmiştir. Bu durumun bir sonucu olarak ortaya yeni ve hızlı büyüyen bir market ortaya çıkmıştır.

Mobil platformlar ve bunların oluşturdukları yeni pazar diğer pazarlara nazaran daha hızlı büyüme ivmesine sahipolmuştur. Fakat budurum beraberinden birçok farklı sorununda ortaya çıkmasına neden olmuştur. Bu sorunlardan en önemlileri arasında sayılabilecek kullanıcı kişisel veri güvenliğidir. Mobil güvenlik terimi, günümüz yeni teknolojilerinin bir sonucu olan mobil platfromlar için kapsamı çok geniş olan bir terimdir. Bu açıdan bu çalışmada kastedilen güvenli sorunu mobil platformlarda kullanıcıların kişisel bilgi güvenliğinin sağlanması ile sınırlıdır.

Bu çalışmada kullanıcı kişisel verileri açısından güvenlik açıklarının sebepleri ve platform bağımsız tüm uygulamalar için geçerli olabilecek bir çözüm önerisi olarak PIPSU(Personel Security For User Private Data) ortaya konulmuştur. Önerilen çözüm PIPSU ile mobil platformlarda kişisel bilgilerin kullanıcı kontrolü olmaksızın kullanılmasının önüne geçilmesi hedeflenmiştir.

PIPSU'nun önemli özellklerini ana başlıklar halinde sıralamak gerekirse:

i. Kullanıcı kişisel verileri için veri türünden bağımsız güvenlik çözümü sağlamaktadır.
ii. Bu çözümü mobil platformdan bağımsız olarak sağlamaktadır.
iii. Bulut bilişim uyumludur. (Cloud Computing Compatible)
iv. SyncML protokol standardı ile farklı bulut sistemleriyle çalışabilmesi mümkündür.
v. Sunum, servis ve depolama katmanları ile uçtan uça bir güvenlik çözümüdür.
vi. Çevrim dışı kipine özel güvenlik ve depolama işlevi barındırmaktadır.

Yukarıda saydığımız özellikleri aracılığı ile PIPSU, kullanıcı kişisel verilerini mobil platform bağımsız olarak saklayabilen ve kullanıcı kişisel veri devamlılığını sağlamaya yönelik bir çözüm önerisidir. PIPSU'nun en önemli artılarından biri, getirdiği yüksek güvenlik seviyesinin kullanıcıların mobil aygıtları ile gündelik kullanım alışkanlıklarını ve etkileşim kiplerini değiştirmek zorunda kalmadan etkili kılmasıdır. Bu durum getirilen önerinin önemli bir güvenlik çözümü olmasının dışında pratik ve ticari uygulanabilirliği açısından önem arzetmektedir.

1.1. PROBLEM TANIMI

Günümüz mobil aygıt kullanıcıları, sahip oldukları aygıtlara seçici olmadanuygulama indirmeye ve kullanmaya başlamışlardır.Bu uygulamaların arka planda çalıştıkları platform ile etkileşimleri, kullanıcı verilerininnasıl kullanıldığı genel olarak son kullanıcı tarafından bilinmemektedir. Mobil platform uygulamalarının gündelik yaşam içerisinde detaylı incelemeden indiriliyor olması, uygulamaların sistem kaynaklarına erişim mekanizmalarının iyi ayarlanamaması,uygulama arayüzlerinin kullanışsız oluşu, uygulamaların kurulum sırasında sistem kaynaklarının kullanımına dair edindikleri hakların çalışma zamanında kullanıcının kontrolünde olmayışı gibi birçok etmen mobil aygıt veri güvenliği sorunlarını beraberinde getirmiştir. Bu uygulamalarla birlikte kurulan kötü amaçlı yazılım ve virüslerin de yaygınlaşmasını artmaktadır [10]. Tüm bunların ışığında mobil işletim sistemlerindeki güvenlik önlemlerinin önemi artmaya başlamıştır.

Uygulamaların karmaşıklığı, API'ler ile erişebilme metodlarının çokluğu, erişilebilecek kaynakların çeşitliliği (kontak, sms, internet erişimi, sosyal medya bilgileri, kullanıcının internetde var olduğu bütün her yere erişim), erişim hakları atanmasının kullanışsızlığı (sadece yükleme zamanında belirlenmesi) gibi etmenler yüzünden kötü amaçlı yazılımların kontrolü tam istenilen gibi olamamaktadır. Bu açıklardan faydalanan kötü amaçlı ve/veya hatalı yazılımlar kullanıcıların kişisel bilgilerini (mobil aygıt lokasyonu gibi) ve verilerini kullanıcı kontrolü dışında elde edip kullanmaktadırlar.

Uygulama geliştiricileri popüler uygulama marketlerine çok cüzi ödemelerde bulunarak, geliştirdikleri uygulamalarını bu marketler aracılığı ile kullanıcılara ücretli yada ücretsiz olarak ulaştırabilmektedirler. Her nekadar Android gibi popüler mobil platfromlar ve bunların sahip oluduğu marketler, kendi platfromları için SDK(Software Development Kit) sağlayarak, geliştirilen uygulamaların çalışacağı platform'a yönelik geliştirilmesini sağlamaya çalışsalar da marketler aracılığı ile kullanıcılara ulaştırılmak istenen her uygulamanın kullanıcı indirmeden önce marketlerce detaylı bir analize ve test edilmesi konusunda başarılı olunamamaktadır. [8], [9]

Mobil platformlarda uygulamaların çok değişik kaynaklar tarafından geliştiriliyor olması, tam güvenilir olmaması ve kullanış kolaylığı açısından yüksek erişim hakları ile

mobil aygıt öz kaynaklarına kolay erişebilir olması güvenlik problemini daha içinden çıkılamaz bir hale getirmektedir.

Google tarafından geliştirilen açık kaynak kodlu bir işletim sistemi olan Android[5] yaygın olarak kullanılan bir mobil platformdur. Android, sistem mimarisi olarak linux tabanlı bir işletimsistemi olmakla beraber Android'i bir linux sürümü olarak tanımlanmaktam doğru olmayacaktır. Android, Linux tabanlı diğer işletim sistemleri gibi Linux çekirdeğiüstüne kurgulanmış olmakla beraber diğer üst seviye kısımları için kendi yapısına özgü bir mimarisi vardır.Google firması tarafından başlatılan ve yönetilen açık kaynaklı bir işletim sistemi olması, Android platformuna hız ve zaman konusunda büyük kazamanımlar sağlamıştır. Android işletim sistemi, sahip olduğu kısa geliştirilme geçmişine rağmen içerisinde bir çok fonksiyonu barındarmaktadır. Bu fonksiyonların geliştirilmesinde görev alan yazılım geliştiricilerin farklı şirketlerden oluşu, farklı yazılım geliştirme disiplinlerine sahip oluşları, yaptıkları çalışmaları gönüllülük çerçevesinde yada çalıştıkları projelere yönelik olarak geliştirmeler oluşu beraberinde bir çok problemi de getirmiştir. Bu problemler arasında:

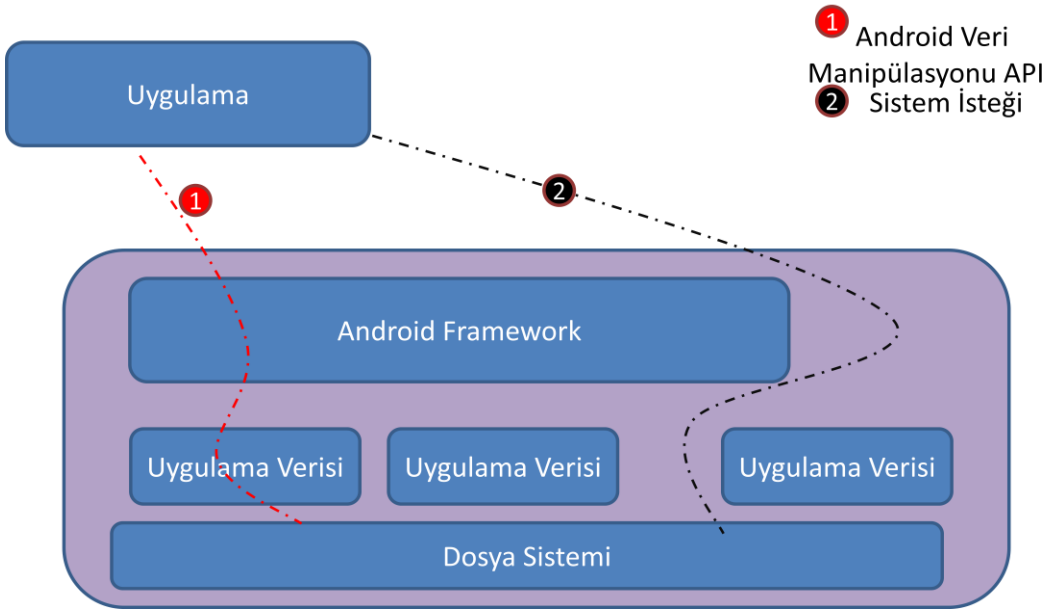
(i) Android mobil platfromunda standardizasyon eksikliği,
(ii) Aynı amaca hizmet eden modüllerin artması,
(iii) Platfrom mimarisinin ve API'nın Android versiyonları arasında büyük oranda farklılaşması,
(iv) Android geliştirmesi üzerinde ortaklaşa çalışan farklı firmaların aynı versiyon numarası altında lanse edilen Android dağıtımlarının birbirinden çok farklı olması,
sayılabilir.

Açık kaynaklı bir proje olarak başlıyan Android mobil platformunun gün geçtikçe Google, Samsun, HTC vb. firmaların açık kaynak olmayan modüllerine bağımlı geliştirilmesi sonucu Android platfromu standartlaşamaması Android'in gönüllü yazılım geliştirmecilerin desteğinin önünü kapayan en büyük engellerin sebeplerindendir.

Bu saydığımız etmenler sonucu Android halihazırda standartlaşamamış olup uygulama geliştiriciler tüm güncel Android versiyonlarını kapsayacak şekilde geliştirme yapmak için büyük efor göstermektedir. Geliştirme zamanı ve maliyetibu durumdan ciddi

oranda olumsuz etkilenmektedir. Android platform geliştiricilerinin birbirlerinden bağımsız çalışmaları sonucu ortaya çıkan aynı adı taşıyan Android platform versiyonları arasındaki tutarsızlık bu platfromlarda güvenlik açıkları ve uygulama taşınabilirliği açısından büyük tehlike teşkileder olmuştur. Ayrıca Android sahip olduğu linux çekirdek yapısını efektif şekilde kullanamamaktadır ve bu durum bir çok engellenebilecek güvenlik açığının ticari sürümlerde yer almasını sağlamaktadır. Akademik çevrelerde, masaütü Linux dağıtimlarını sahip olduğu standart SeLinux gibi güvenlik yönetim modüllerinin Androidişletim sistemine entegre edilmesi gerektiği tartışılmaktadır [6].

Kişisel bilgisayarların yerini almaya başlıyan bu platformlar gittikçe artan miktarda ve önemde kullanıcı kişisel bilgilerini bulundurmaktadır. Fakat yazılım ve donanım mimarisi henüz tam olarak standartlaşamayan bu platformlar güvenlik gereksinimleri yönüyle yeterince olgunlaşamamış ve bunun sonucu olarak kullanıcı kişisel bilgilerinin korunması için gerekli altyapıyı sağlayamamaktadırlar [11,12]. Ayrıca günümüz mobil platformların değişen mimari yapıları, sağladıkları yazılımsal ve donanımsal yeniliklerin bir sonucu olarak mobil uygulamaların yetenekleri ve çalışma tarzları sadece basit telefon temelli uygulamalardan, çok işlemli (multi threaded) servis modeline doğru kompleksleşen bir yapıya doğru gitmektedir. Bu yeni yapıda uygulamalar platform API'lerini kullanarak sistem kaynaklarını kullanabilmekte ve mobil aygıt'ın hemen hemen tüm kaynak ve olanaklarına teorik olarak erişebilmektedir. Mobil uygulamaların yeteneklerinin artması, mobil platformların mobil uygulamalar tarafından erişilebilen API lerinin ve sistem kaynaklarının güvensiz olması sonucu, mobil platform kullanıcısına ait kişisel bilgilerin ilgili kullanıcı kontrolü dışında mobil uygulamalarca internet ortamına taşınması ve kullanılması gibi ciddi güvenlik açıklarını ortaya çıkardı.



Şekil 1.2Android Uygulama Veri Kullanım Modeli

Şekil 1. 2 de günümüz mobil platformlarında bulunan kullanıcı kişisel bilgilerinin mobil uygulamalar tarafından kullanımı gösterilmeye çalışılmıştır. Mobil uygulamalar ve bu uygulamaların mobil platformlar üzerinde bulunan kullanıcı verilerine erişimleri iki ana yol ile mümkünolabilmektedir.

- Mobil Platform API
- Mobil Platform Dosya Sistemi

Günümüz mobil platformlarında veriler aşağıda sıralanan konumlarda saklanabilmektedirler;

- Kavramsal Konum:
  - Uygulamanın kendi veritabanı(SQLite DB)
  - Android Shared Preference
  - PIM veritabanı(Telefon Rehberi, SMS, Takvim vb.)
- Fiziksel Konum:
  - Dahili Veri Alanları
    - Flash Memory (SSD)
    - EMMC
    - Anakartın içsel ROM'u
  - Harici Veri Alanları

- SD Card
- Ağ Bağlatısı Erişimli Veri Alanı(NAS: Network Attached Storage)

Günümüz mobil platformlarında bulunan güvenlik açıklarını özetlemek ve sınıflandırmak gerekirse;

- Mobil platform sağlayıcıları tarafından önlemi alınmayan işletim sistemi açıkları: Mobil işletim sistemi sağlayıcıları geliştirdikleri mobil platformlara, onların dışındaki geliştiriciler aracılığı ile de uygulama geliştirilebilmesi amacıyla iletişim arabirimleri ve API ler tasarmaktadırlar. Bu arabirimler kötü amaçlı mobil uygulamalarca veya bu aygıta bağlanmayı sağlayan masta üstü uygulamalarınca kullanılabilmektedir. Günümüzde bu sorunların aktif olarak yaşandığı mobil platformlardan biri de Androidtir. Android sahip olduğu ADB (Android Debug Bridge) iletişim arabirimi ile uygulama geliştiricilere ve masaüstü uygulamalarına bir ileştişim birimi ve API sağlamaktadır. Android telefon veya tabletlerin barındırdıkları kullanıcı kişisel bilgilerini ilgili cihaz dışında yedeklemeyi sağlayan bir çok masaüstü uygulaması bulunmaktadır. Bu uygulamalar ADB aracılığı ile kullanıcı kişisel bilgilerini kopyalayabilmekte ve hatta cihaz üzerinden silebilmektedir. ADB Android sistemi içerisinde bir çok şeyi yapabilme kabiliyetine sahip olduğu için, uygulama geliştiriciler Android işletim sistemine sahip olan mobil cihazlardan ADB aracılığı ile;
    - Kullanıcı PIM (Personal Information Manager) bilgilerini yani sms, kontak, takvim, not vb bilgileri
    - Cihaz içerisinde ADB ye erişim hakkı tanınan dahili ve harici disk arabirimlerine erişerek buradaki verileri
    - Kullanıcı multi- media verilerini

  kopyalayabilmekte, silebilmekte veya degistirebilmektedir.
- Mobil platformların sahip oldukları erişim hiyerarşi (access policy) açıkları: Mobil platformlar, örneğin Android sahip olduğu çekirdek (kernel) ile bir dosya sistemi ve bu dosya sistemine bağlı olarakta birden fazla katmanda bir erişim hiyerarşisi oluşturmaktadır. Fakat bu hiyerarşi ilgili platform için geliştirilen farklı fonksiyonların implementasyonunda güvenlik açıklarının çıkmasına neden olmaktadır. Örneğin Android işletim sistemi sahip olduğu Android Framework

arabirimi ile kullanıcıların kişisel bilgileri SQLite veri tabanlarından saklamaktadır. Bu veri tabanlarının kullanımıAndroid Framework API üzerinden ön tanımlı erişim hakları ile sağlanabilmektedir. Fakat bu veritabanlarının fiziksel dosya sisteminde bulunduğu dizin ve bu dizine erişim bir çok Android versiyonunda farklı fonksiyonlar için implemente edilen APIlerce kontrol dışı sağlanmaktadır. Dolayısıyla kullama hakkına sahipolmayan uygulamalar bu ortamda bulunan kullanıcı kişisel bilgilerini bu veritabanların bulundugu dosya sistemine erişerek cihaz dışına taşıyabilmektedirler.

- Uygulama yetkilerinden kaynaklı kişisel veri güvenlik açıkları: Android gibi mobil platformlar, mobil uygulamaların kurulum ve çalışma zamanında yapacakları işleri kontrol altına alabilmek için yetkilendirme hiyerarşisi oluşturmaya çalışmaktadırlar. Yetkilendirme hiyerarşisi genelde kurulma zamanında belirlenen ve çalışma zamanında değiştirilemeyen bir mekanizmaya sahip olan platformlar için soruna neden olmaktadır. Örneğin kötü amaçlı bir Android mobil uygulaması düşünelim ki tek görevi cihaz üzerindeki kameranın sahip olduğu flash ışığını açıp kapatabilen bir fener uygulaması olsun. Bu uygulama kurulma zamanında cihaz kullanıcısına yapacağı işlere dair izinleri sormaktadır. Cihaz kullanıcısı izinler arasında eleme yapamadığı için ilgili uygulamayı kurabilmek için uygulamanın kullanıcıdan onaylamasını istediği bütün izinlerini onaylamak zorundadır. Bütün izinleri alan bir fener uygulamasınınçalışma zamanındaki davranışı değiştirilemediğinden kullanıcının kontrolü olmadan cihaz üzerinden kullanıcı kişisel verilerini kullanabilmektedir.Ayrıca birkez sistem öz kaynaklarına erişim hakkı tanınan uygulama daha sonra uygulama güncellenmesi,internet, sms gibi birçok yötem ile ilgili platform'a trojan veya virüs bulaştırabilir ve kişisel bilgilerimizi bizim kontrolümüz dışında mobil cihazımız dışına taşıyabilir. Günümüz mobil platformlarının temel kullanım amaçlarını kısaca sıralamak gerekirse;
    - Telefon, E-Posta, SMS, Chat gibi uygulamalar aracılığıyla diğer insanlar ile iletişime geçebilmek.
    - Sosyal ağ uygulamaları, haber servisleri, RSS vb uygulamalar ile internet ortamını yakından takip edebilmek ve bağlatılı akalabilmek

- Lokasyon(GPS) tabanlı servisler aracılığı ile navigasyon ve sosyal ağları kullanabilmek
- Mobil web tarayıcıları aracılığı ile internette gezinebilmek
- Mevcut ticari yazılımlara mobil platformlar aracılığı ile heryerden daha kolay uylaşabilme ve kullanabilme
- Sağlık,Finans veya Haber gibi kritik bilgilere anında ulaşabilmek.

Yukarıdaki kullanım amaçları göz önünde bulundurulduğunda ortaya çıkan güvenlik sorunu ciddi boyutlara sahip olmaktadır.

1.2. ÇÖZÜM ÖNERİLERİ VE LİTERATÜR TARAMASI

Android platform'u nun da diğer mobil platformlarda varolan temel güvenlik açıkları olduğunu bir çok akademik kurum ve ticari şirketler bilmekte ,bu sorunu giderebilmek amacı ile çok yönlü çalışmalar yapmaktadırlar. Yoğunluklu olarak ticari çevreler tarafından geliştirilmesi süren android platformunun sahip oluduğu arakatman(Middleware), güvenlik açısından bir çok açığa sahip olup ve bu açıklar aracılığı ile Soundcomber gibi trojan veya virüs saldırılarının Android arakatmanı tarafından fark edilemediği bir çok çalışma ile ortaya konulmuştur.

Temel kullanım amaçları göz önünde bulundurulunca, mobil platformların hayatımızdaki yerleri de gün geçtikçe önem kazanmakta olup, kullanıcıların bu platformlar üzerinden yaptıkları işlemlerde güvenli olmak zorundadır. Fakat Android başta olmak üzere bir çok mobil platform üzerinde kullandığımız sosyal ağ hesaplarımız, telefon rehber bilgilerimiz, mail hesaplarımız, bulunduğumuz konum bilgisi ve daha bir çok kullanıcı kişisel bilgileri kontrolümüz dışında mobil platformlar ve servisler aracılığı ile internet ortamına ulaştırılabiliniyor.

Mobil aygıtlar ve özellikle Android güvenliği ile ilgili birçok çalışma bulunmaktadır. Bu çalışmaların bir kısmı uygulama altı işletim sistemi seviyesi güvenliğine odaklanmaktadır [19, 20]. Ayrıca Android işletim sistemi dosya güvenliği, telsiz haberleşme şifrelemesi gibi konularda da çalışmalar bulunmaktadır [20]. Bu çalışmanın

esas konusu olan uygulama seviyesi güvenliği ile ilgili geliştirilen çözümleri şu ana başlıklarda toplamak mümkündür:
İncelediğimiz çalışmaları önerdikleri güvenlik çözümlerini açısından 3 grupta toplamaya çalıştık;

1.2.1. YETKİ ETİKETLERİ TEMEL ALINARAK YAPILAN ÇALIŞMALAR

Kirin [15,16,17] mantıksal bir uygulama olan bu çalışma,Android mobil platfromunun global güvenlik yetki etkiteleri ve bunların platfrom üzerinde neden olabilecekleri güvenlik açıklarını baz alarak, Android platfromuna kurulacak uygulamaların yetki etiketleri üzerinden uygulamanın kuruluma zamanında kullanıcıları bilgilendirmeye yönelik önerilen bir çözümdür. Kirin yaptığı analiz sonucu Android platfromuna kurulacak uygulamaları tehlikeli veya güvenli gibi sınıflara ayarmaktadır.

Semantically Rich Application-Centric Security in Android(Saint) yine Android uygulamalarının kurulma zamanı bilgi ve yetki etiketlerinin tanımlandığı AndroidManifest.xml üzerinden kurulma zamanına bağlı olarak uygulamaların yapabilcekleri üzerinden uyarılarda bulunan arac, çalışma zamanında çözüm olarak yetersiz kalmaktadır.

1.2.2. ÇALIŞMA ZAMANI UYGULAMA TAKİBİNE DAİR YAPILAN ÇALIŞMALAR

Language based security on Android[13] ;Android uygulamalarında uçtan uca bir güvenlik çözümü sunabilmek için Android mobil platfromu'na özel veri tiplerinden oluşan bir programlama dili önerisinde bulunmaktadır. Bu öneri modeli ile Android uygulamalarının temelde göz önünde bulundurmaları gereken güvenlik etmelerini implementasyon düzeyinde temel birimler olarak implemente etmeyi hedeflemektedir. Böylece, Android uygulamaları temelde dikkat etmeleri gereken güvenlik önlemlerini bu modelin önerdiği dil ve bu dilin sunduğu kütüphaneler aracılığı ile disipline etmeyi hedeflemektedir.

SCanDroid[14] ise Language based security on Android'e benzer bir modeli veri tipleri tasarlamak yerine veri akışlarını izleyerek sağlamayı hedeflemektedir. SCanDroid veri

tiplerinin standardizasyon açısından önemli olmasının yanında veri akışlarını izeleyerek ve bunlara bağımlı çalışma zamanında geçerli olabilecek bir güvenlik önerisinde bulunmanın implementasyon açısından daha yararlı olacağını savunmaktadır.Bu model önerdiği veri akış izleme güvenlik çözümünün türden bağımsız yani hem platform tarafında ön tanımlı veri tiplerinin akışlarının takibinde, hemde kullanıcı tanımlı veri tiplerinin takibinde aktif olarak kullanılabileceğini savunmaktadır.

1.2.3. AĞ TABANLI YAPILAN ÇALIŞMALAR

Virtualized In-Cloud Security Services for Mobile Devices [18], Mobil aygıtlar için antivirüs fonksiyonalitesini taşımayı hedefleyen model, alışılmış mobil antivirüs çözümlerinin aksine, çözümü bir mobil işletim sistemi uygulaması olarak geliştirmek yerine, ağ tabanlı kötü amaçlı yazılım saptayıcı bir ağ yazılımları dizisi olarak tasarmıştır. Çözüm mobil aygıtlarda kötü amaçlı yazılımların saptanması için harcanan bellek, güç ve işlemci tüketiminin, Virtualized In-Cloud Security Services for Mobile Devices'ın sağladığı ağ tabanlı kötü amaçlı yazılım belirleme servisleri aracılığı ile minimuma ineceğini ve mobil cihazlardaki ortam kısıtları sonucu yaşanan sorunların ağ tabanlı çözümleri ile ortadan kaldırılacağını savunmaktadır. İlgili model, her nekadar mobil aygıtlardaki donanım veya yazılım bağımlı kısıtlardan kaynaklı harcanan bellek güç ve işlemci tüketimini minimuma indireceğini savunsa de, mobil aygıtların ağ'a bağılı olmadıkları ortamlar, internet gibi açık ortamlar veya şifrelenmiş protokol kullanımları gibi durumlarda başarısız olamaktadır

3 Ayrı sınıflandırma yapılarak incelenen bu çözümler, Android uygulamalarının genel güvenlik açıkları ile ilgilenmektedirler. PIPSU çözümü sadece Android uygulama güvenliğini incelemeyip detayda kullanıcı kişisel bilgileri ve bunların mobil Android aygıtlarında kullanımı(Import/Export, Contacts, Calendar, vb) sırasındaki sorunuları inceleyip bunlara dair çözüm üretmeyi hedeflemektedir. PIPSU ile benzer çalışmalar yapan modelleri ve bunların çözüm önerileri açıklanmaya çalışılmıştır. Farklı akademik kurumlarca yapılan bu çalışmalar, Android mobil platformunun güvenlik açıklarından kaynaklı farklı sorunları çözmeyi veya iyileştirmeyi hedeflemektedirler.Fakat bu modellerin bir çoğu sundukları çözüm önerileri ile genel Android uygulamalarının kurulum ve çalışma zamanındaki davranışları ve bu uygulamalar aracılığı ile yada bu uygulamalara yapılan saldırıları takip etmeye ve önlemeyi hedeflemektedirler. Yani

genelllikle bu çözümler Android uygulamalarının kurulma zamanı ile ilgili güvenlik çözümleri veya çalışma zananında Intent ve Broadcast gibi android IPC(Intern Process Communication)/ ICC(Internal Component Communication) metodları ile haberleşme eventlerini takip etmeye yönelik hazırlanılmış çözümlerdir. Fakat hiç biri detayda mobil platform kullanıcısının kişisel bilgileri ve bunlara Android ara katmanında erişmeyi sağlayan ContentProviderlar ve bunların saklandığı SQLlite veritabanlarının güvenliği veya kullanımını detayda incelememektedirler. Bu yönü ile kullanıcı kişisel bilgilerinin sadece Import/Export işlemlerinde Android mobil platform'u ile etkileşen fakat günlük kullanımında mobil Android platfromu üzerinde kullanıcı kişisel bilgilerine dair hiç bir iz bırakmayıp, Android mobil platformunun kullanıcı davranışını değiştirmeyerek kullanıcı alışkanlıklarında bir kolaylık sağlayan iki uçlu bir çözümdür. Burada "iki uçlu" benzetmesi, bir ucu mobil platformda diger ucu ise birden fazla mobil istemciye güvenlişekilde hizmet vermeyi hedefleyen bulut bilişim yaklaşımlarıyla geliştirilecek bir servis dizisidir. PIPSU çözümü önerdiği içsel olmayan Android uygulama modeli ile incelenen diğer çalışmalara göre çok daha fazla ortam bağımsız bir çözüm olabilmektedir.

1.3. TEZİN AŞAMALARI

Sorunun açık bir şekilde ortaya konabilmesi için sorunun kaynakladığı platformolan Android ve Android tabanlı teknolojilerin detaylıca incelenmesi gerekmektedir. Android platformu ve bu platformda bulunan teknolojiler detaylıca analizi, Android platformunda bulunan kullanıcı kişisel veri güvenliğine dair sorunların belirlenmesini ve üreteceğimiz çözümlerin daha bilimsel bir zemine taşınmasını sağlayacaktır.

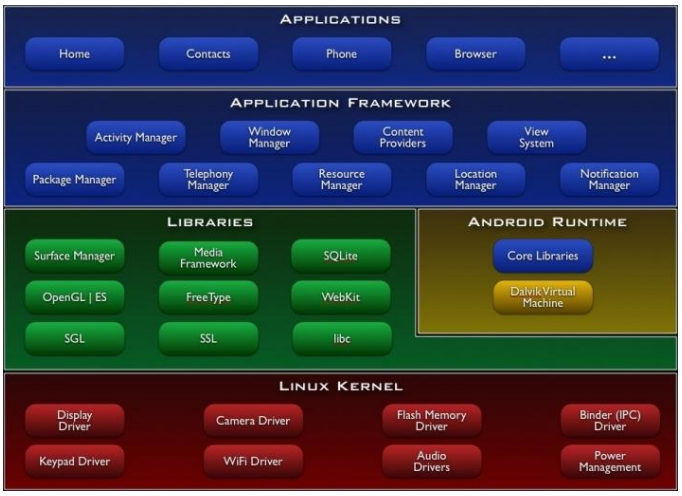
Android platform analizinin ardindan PIPSU çözüm platformu hem teknik yapısı hemde kullanım modeliyle ortaya konulmaya çalışılmıştır. Bu Tez çalışmasında PIPSU platform'u iki ana bölüme ve bu bölümlerde kullanıcı açışısından ve teknik açıdan ele alınmaya çalışılmıştır. Bu bölümler sırasıyla;

- İstemci tarafı olan PIPSU-L'nin diğer Android uygulamaları ile karşılaştırılması.
- PIPSU-L kullanıcı kişisel bilgi soyutması ve modellenmesi.
- PIPSU-L'nin teknik mimarisi ve istemci tarafta sunduğu güvenlik modeli.
- Bulut tarafı olan PIPSU-C'nin teknik mimarisi ve iletişim altaypısı.

Yararlanılan açık protokoller ve kaynak kodlu projeler anlatılmaya çalışılmıştır.

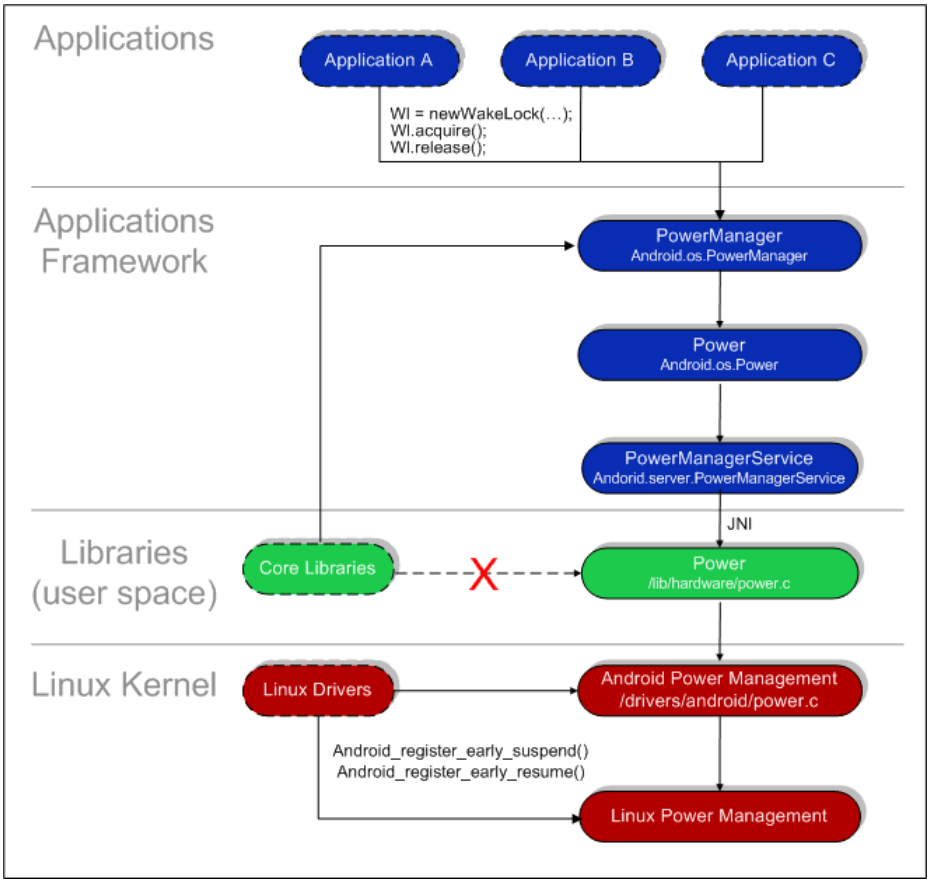
2. MOBİL İŞLETİM SİSTEMİ OLARAK ANDROID MİMARİSİ

Bu bölümde Android'in genel sistem mimarisi ve sahip olduğu güvenlik altyapısını incelemeye çalışılmıştır.



Şekil2.1 Android Platform Mimarisi

Açık kaynak kodlu mobil bir platform olan Android bir linux çekirdeği, bir Android ara katmanı ve birde uygulama katmanınından (Application Framework) oluşmaktadır (Şekil 2.1) [20]. Temelde bir linux çekirdeği üzerinde katmalı olarak mimarilendirilmiş, temel ortam arabirimlerini yönetmeye ve bunlara erişmeye yönelik tasarlanmış alt seviye kütüphaneler den oluşan bir katman ve bu katmanla aynı seviyede bulunan temel amacı; sistem API'ını oluşturmak ve Java tabanlı android uygulamalarını çalıştırmak ve yaşam döngülerini düzenlemektir (Life-cycle).



Şekil2.2 Katmanlı Android Mimarisi

Şekil 2.2 Android mimarisini katmansal olarak göstermektedir [21]. Temelde iki tür Android uygulamalaması bulunmaktadır. Bu iki uygulama türü, Android'in uygulama katmanına erişim yetkileri ve uygulama katmanını kullanımları açısından farklılaşmaktadır.

- İçsel Android (Built-in) Uygulamaları
- İçsel Olmayan (Non-Built-in) Android Uygulamaları

Android uygulama katmanı içerisinde, Android işletim sisteminde kullanılmak üzere iki tür sistem API'si bulunur.

- Birincil tür Android işletim sisteminde çalışma zamanında kullanılabilinen ve Android işletim sistmenide bulunan her türlü Android uygulmasının erişebildiği bir API türüdür
- İkincil API türü ise, Android işletim sisteminin derleme zamanında Android işletim sistemi ile derlenen ve Android uygulama katmanında sadece İçsel

Android uygulaması türündeki(işletim sistemi ile derlenen ve çalışma zamanında kaldırılamayan) uygulamaların kullanımına ve erişimine izin verilmiş sistem API'ı dır.

2.1. İÇSEL ANDROID UYGULAMALARI

Bu tip uygulamalar, Android tarafından yüksek güvenlik önlemi ve optimum sistem kaynağı tüketimi gerektiren işlevlerin yerine getirilmesini hedefleyen uygulamalar olarak tasarlanırlar. İçsel Android uygulaması uygulamalar Android işletim sisteminde özerk farklı kullanıcı gruplarına atanarak, sistemde bu uygulamaların eriştiği sistem kaynakları güvenliği sağlanmış olur. İçsel Android uygulamaların içerisinde bulunduğu kullanıcı grupları sayısı arttırılarak İçsel Android uygulamaları bir birlerinden izole edilirler. İçsel Android uygulamaları çalışma zamanında kaldırılamazlar veya güncellenemezler. Bu uygulamaları değiştirmek yada kaldırmak için Android işletim sistemi ile aynı işarete(Signature) sahip bir güncelleme ile istenilen işlemi yapmak mümkündür.

2.2. İÇSEL OLMAYAN ANDROID UYGULAMALARI

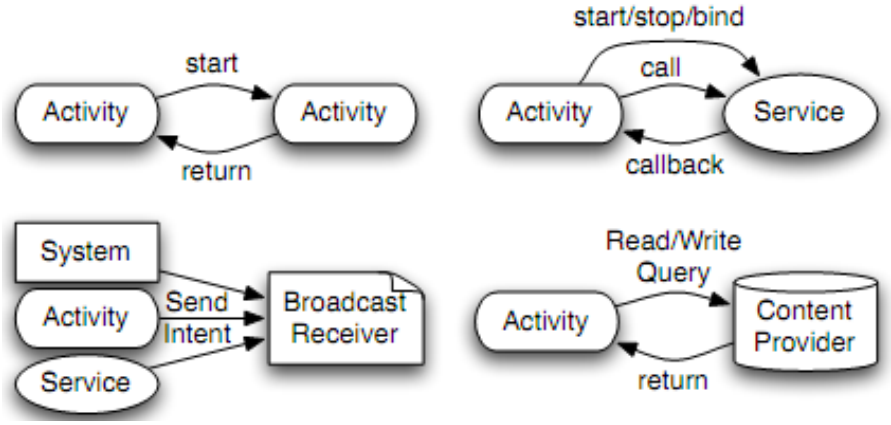
Bu uygulamalar ise Android çalışma zamanında kullanıcılar tarafından yüklenebilen ve istenildiği taktirde kaldırılabilen standart Android uygulamalarıdır. Android'in uygulama geliştiricilere sağladığı SDK(Software Development Kit) ve IDE(Integrated Development Environment) plugin'leri aracılığı ile geliştirilebilinen bu uygulamalar Android işletim sistemine özel bir paketleme yapısı olan APK(Android Package) olarak pakatlenerek Android işletim sistemine kurulabilir hale getirilirler.

Android Çalışma Zamanı(Runtime)katmanı,bu katmanların üzerinde ise işlevsel kütüphaneler topluluğu(Framework) ve bunlar aracılığı ile sağlanan İçsel Android uygulamaları yada içsel olamayan Android sistem uygulamalarına API düzeyinde servis sağlayan katman vardır. Mimarinin en üst katmanı ise uygulamalar ve launcherlardan oluşmaktadır. Temelde android uygulamaları Java programlama dili ile android tarafından sağlanan SDK ile geliştirilip byte kod olarak derlenirler ve android üzerinde

bulunan Dalvik sanal makinasında yorumlanırlar[7]. Uygulamaların system ve diğer uygulamalar ile iletişimi (IPC) middleware'de bulunan "binder IPC" aracılığı ile sağlanır. Android uygulamaları temelde bileşenlerden oluşur ve bu bileşenler android mimarisinde bulunan intent öğeleri aracılığı ile haberleşebilirler.

Android Intent Bileşenleri: Intent'ler Android işletim sisteminde, Android uygulamaları arası veya bir Android uygulaması ile Android işletim sistemi arasındaki asenkron mesajlaşmayı sağlayan mekanizmalardır. Android Intent'ler bir verinin bir Android Activity'si yada Android servisine gönderilmesi yada ondan alınmasını sağlayan mekanizmalardır. Intent'ler aracılığı ile Android işletim sisteminde birbirinden bağımsız geliştirilen uygulamaların veya işletim sistemi bileşenlerinin mesajlaşmaları sağlanmaktadır.

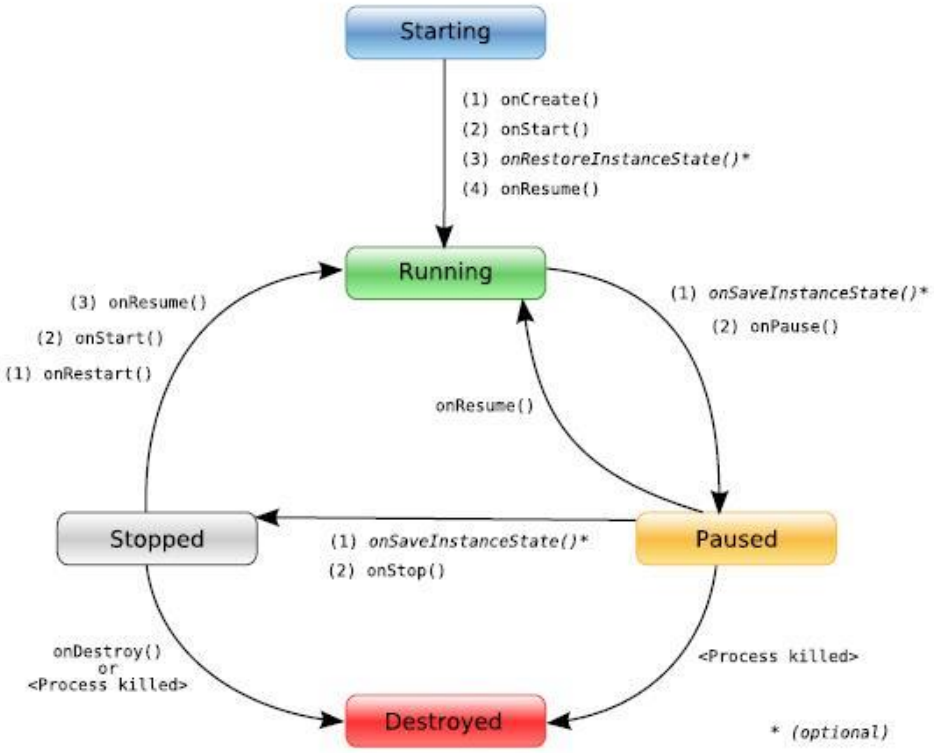
Android uygulama mimarisinde temelde dört çeşit component vardır:



Şekil2.3 Android Uygulama Bileşenleri

Bunlar sırasıyla;

Activity: Uygulamanın kullanıcı ara birimi olup dokunmatik ekran veya tuş takımı aracılığı ile bu arayüzlere kullanıcı verisi yazılıp okunabilinir. Activitylerin android mimarisindeki yapıları gereği çalışma zamanı içersisinde sadece bir activity çalışabilir/görünebilir. Bu sırada daha önce geri planda çalıştırılmış activityler pasif duruma taşınır.

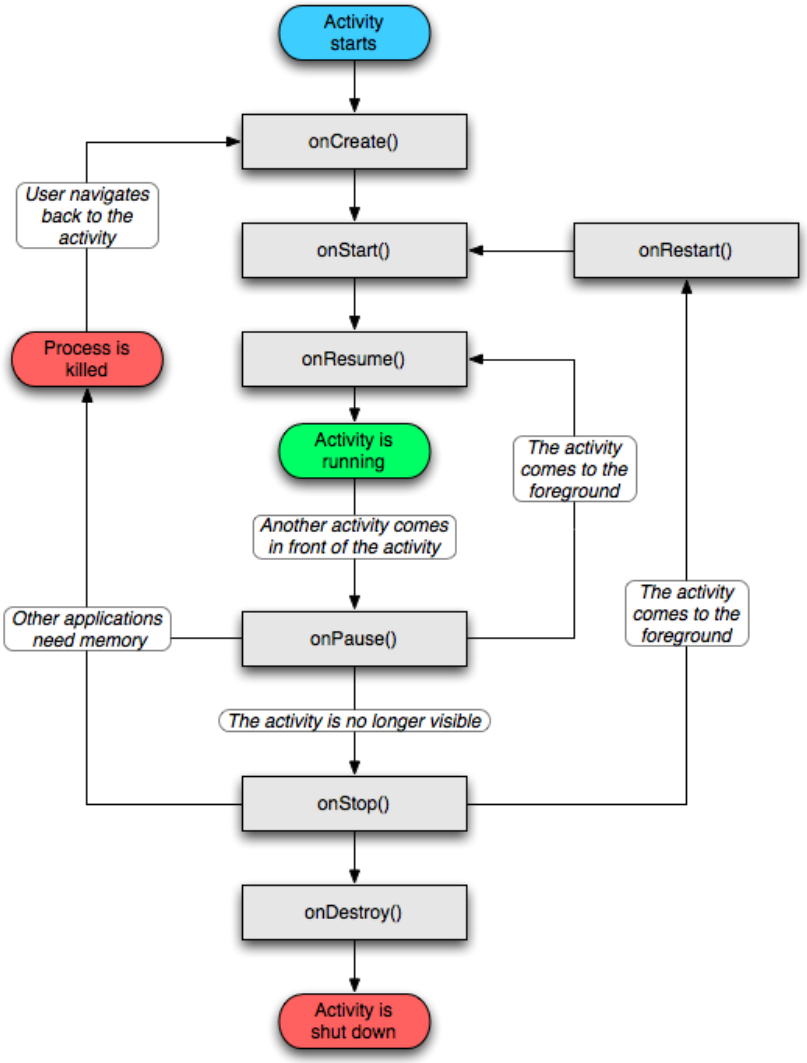


Şekil2.4 Bir Android Activity'sininYaşamDöngüsü

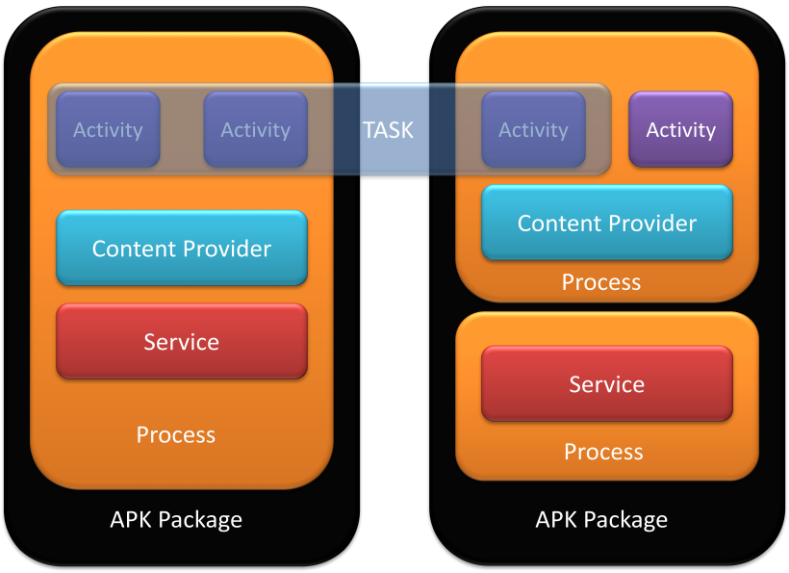
Service: Servisler uygulamanın arka planda yapması gereken işlerini Activity bağımsız yapmalarını sağlarlar. Servisler ayrıca RPC(Remote Procedure Call) arayüzleri sayesinde takip edilinebilinir ve servislerin ürettikleri veya servislere beslenmesi gereken event ve parametereler yaratılıp kullanılabilinir.

Broadcast Receiver: Bu tipteki bileşenler ise sistemce üretilen asenkron event (İnternet bağlantı durumunun değişmesi, SMS veya Telefon aramalarının takibi vb.)takibi için kullanılırlar.

Content Providers : Bu tip bileşenler ise android sistemi veya bir uygulama tarafından bütün sistemce erişilebilinen kaynaklara erişim sağlayan arabirimlerdir. Örneğin, Telefon rehberi, system üzerinde bulunan resim, müzik vb. dataların uygulamalarca erişimini sağlayan arabirimlerdir.SQL benzeri erişim arayüzüne sahiptirler.



Şekil2.5 Bir Android Activity'sinin Sahip Olduğu Durumlar



Şekil 2.6 Android Uygulama Bileşenleri

2.3. ÖZEL ANDROID UYGULAMALARI

Launcherlar, Android'in temel mimarilerisinde birer uygulama olup, yine android uygulamalarının temel iki türü olan İçsel ve içsel olmayan Android uyglamaları da launcherlar olarak geliştirilebilirler. Launcherları Android uygulamalarından ayıran temel özellikleri android işletim sistemince sağlanan özel yetenekleridir. Android işletim sistemi çalışabildiği donanımlarda bulunmasını zorunlu kıldığı;donanım temelli tuş takımları ve bunların sistemce üretilen ve hiç bir uygulama tarafından üretilmesine engel olunamayan eventlerini temelde launcherlara yönlendirmektedir. Bu özel evetlerin yönlenmesi sonucu launcherlar her hangi bir çalışma zamanında, sistemin başlangıcında, sistemin kapanışındadonanımın grafik arayüzünde aktif rol oynamaktadırlar. Ayrıca launcherlar sistemce kritik role sahipolan ayarlar menüsü ve sistemde bulunan uygulamarın kısa yollarının eklenmesi ve kısa yolların görünüşünü sağlayan ana aktörlerdir.Launcherlar android tarafından sağlanan iki ana geliştirme kiti aracılığı ile gelitirilebilirler. Bunlar:

- Android Yazılım Geliştirme Kiti(Android Software Development Kit)
- Android Temel Geliştirme Kiti(Android Native Development Kit)

Bütün android uygulamaları gelitirilmesi durumunda android SDK yı kullanırken, laucherlar ayrıca Android NDK yı kullanabilme yetenekleri ve sistemdeki özel davranışlarından dolayı OpenGL gibi system donanımlarını ve kaynaklarını direct olarak kullanabilir ve bunlarla performans gerektiren animasyon ve grafik davranışlarını sağlayabilirler.

2.4. UYGULAMA DÜZEYİ ANDROID GÜVENLİK MODELİ

Bir Android uygulaması sahip olduğu verileri, Android güvenlik modelleri çerçevesinde başka uygulamalar ile paylaşabilir, başka uygulamaların sahip olduğu verilere ,aralarındaki güvelik model hakları dahilinde erişebilir. Android uygulamaları Android'in sahip olduğu Linux çekirdek yapısından dolayı, çalışma zamanında Dalvik sanal makine(Dalvik VM) üzerinde bir Linux UID ile hayata başlarlar ve bütün çalışma hayatları bu kimlik üzerinden takip edilirler.Android uygulamaları Linux uygulamaları gibi Linux çekirdek yapısının sağladığı yönetim ve güvenlik yapı aracılığı ile sahip

oldukları dosya ve veri yapıları ,sadece ilgili Android uygulaması tarafından erişime yetkilidirler.

Temelde Android platformu uygulama düzeyinde üç tür güvenlik modeline sahiptir,

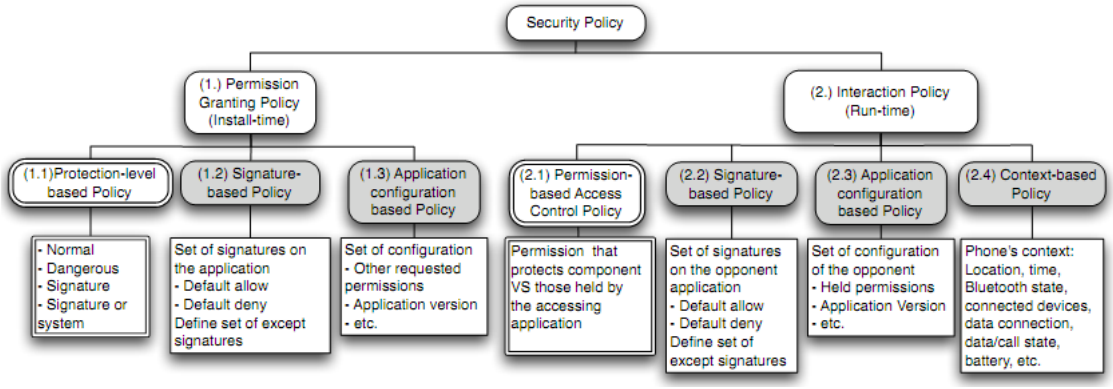
I. Yetki Modeli: Yükleme zamanı yetki belirlenimi
II. Sandbox Modeli: Uygulama kaynaklarının izole edilmesi
III. Uygulama İşaretlemesi: Uygulamalar arası haberleşme ve veri paylaşımı güvenliği

2.4.1. YETKİ ETİKETİ MODELİ

Yetki etiket yapısına sahip olan android güvenlik mimarisi basit ve alışılmış bir yaklaşım ile uygulamaların yetkilendirilmelerini ve böylece uygulama güvenliğini sağlamaktadır. Temelde bir yetki etiketi system veya uygulamalarca tanımlanabilinen ve sistemde tekil(unique) olması gereken bir karakter katarıdır.Android bir çok yetki etiketini system düzeyinde tanımlamıştır. OS-centric bir perspektif ile uygulamaların çalışma zamanında erişecekleri kaynaklar ve yapabilirlikleri kurulum(Installation) zamanında bilinebilmektedir. Bir uygulamaya dair yetki etiketleri ve uygulamanın sistemce bilinmesi gereken bütün ön tanımlı gereksinimleri AndroidManifest.xml olarak uygulama ile birlikte derlenen ve uygulamanın kurulum aşamasında sisteme tanıtılan arayüzü ile tanımlanmış oılunur. Yetki etiketleri sistemce öntanımlı yetki guruplarından birine dahil olmak zorundadırlar. Bunlar temelde

- “normal”
- “dangerous”
- “signature”
- “signature or system”

olarak sınıflandırılmıştırlar.



Şekil 2.4.1.7 Android GüvenlikEtiketHiyerarşisi

2.4.2. ANDROID SANDBOXING MODELİ

Bu model de android uygulamalarının installation zamanında belirledikleri ve çalışma hayatlarına başladıkları güvenlik konfigurasyonları ve bunun sonucunda erişebilecekleri sistem kaynaklarının değişmeyeceğini garanti alatına alan güvenlik modelidir. Yani bir Android uygulaması kurulma zamanında AndroidManifest.xml adındaki konfigurasyon dosyasında belirttiği ve sahip olduğu erişim ve kullanım haklarını değiştiremez. Böylece çalışma zamanında başka bir Android uygulamasına verilmiş yada ayrılmış sistem kaynaklarına ulaşamaz.

2.4.3. UYGULAMAİŞARETLENMESİ

Bu güvenlik modelinde ise çeşitli matematiksel algoritmalar sonucu üretilen bir veya birden fazla güvenlik sertifikası aracılığı ile ilgili uygulama işaretlenir.Sahip olduğu güvenlik algoritması ve bu algoritmaya beslenen parametrik giriş değerleri sadece uygulama geliştirici tarafından bilinen ve barındırılan bir sertifika aracılığı ile uygulama derleme aşamasında işaretlenir. Android ara katmanında bulunan Android Packet Manager olarak bilinen ve sisteme kurulan ve sistemden kaldırılan uygulamalrı Android yöneten paket yönetim modülü, işaretlenmiş uygulamaları kurulum zamanında android sistemice belirler ve bunların öz kaynakları veya bunların kullandıkları

sistemkaynaklarının erişimini güvence altına alır, dolayısıyla işaretlenmiş bir uygulamanın öz kaynaklarına erişebilmenin veya öz kaynaklarını kullanabilmenin tek yolu yine aynı sertifika ile işaretlenmiş başka bir uygulama üzerimden erişmektir.

3. ÖNERİLEN ÇÖZÜM- PIPSU

Bu çalışmada PIPSU hem kavramsal bir tasarım olarak kurgulanmış ve önerilen işlevleri başarıyla bütünleştiren bir mimari yazılım tasarımı olarak tüm detayları ile ortaya konmuştur hem de ortaya çıkan tasarım Android platformu için gerçeklenmiş ve çalışan bir Android uygulaması ortaya çıkarılmıştır.

PIPSU,bulut bilişim tabanlı kullanıcı kişisel veri depolama servisleri ve bu servislerden hizmet alan mobil platform katmanlarından oluşan bir güvenlik platformudur. Bulut bilişim tabanlı katmanı kendi içerisinde servis yönelimli (SOA: Service Oriented Architecture) bir mimariye sahip olup, geliştirilen servisler büyük sayıda mobil platform kullanıcılarının, bu platform servislerinden yararlanmaları hedef alınarak tasarlanılmıştır. Üst düzey mimari olarak, PIPSU platformunda bulunan her servis temelde iki uca sahiptir.

- Bulut Servis Katmanı (PIPSU-L)
- Mobil Servis Katmanı (PIPSU-C)

Bu iki katmanın iletişimi ise SyncML tabanlı mesaj enkapsülasyon protokolü ile güvenli veri transferini sağlayan bir haberleşme katmanı tarafından sağlanmaktadır. PIPSU çözümü üst düzeyde, amaca yönelik tasarlanmış olan şu servis katmanlarından oluşmaktadır:

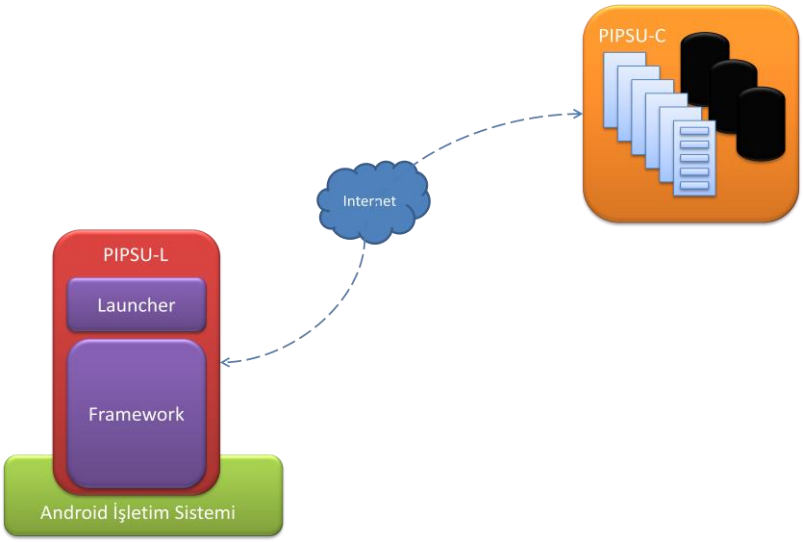
- Kullanım / Sunum Katmanı
- Mobil Platformdan İzole Depolama Katmanı
- Mobil Platform Bağımsız Veri Bütünlük ve Devamlılığı Katmanı

Sayılanbu üç katmandan ilk ikisi PIPSU-L, yani PIPSU mobil istemci tarafında,sonuncusu ise PIPSU-C yani PIPSU bulut servisleri tarafında yer almaktadır.

3.1. PIPSU PLATFORM MODELİ VE ANA BİLEŞENLERİ

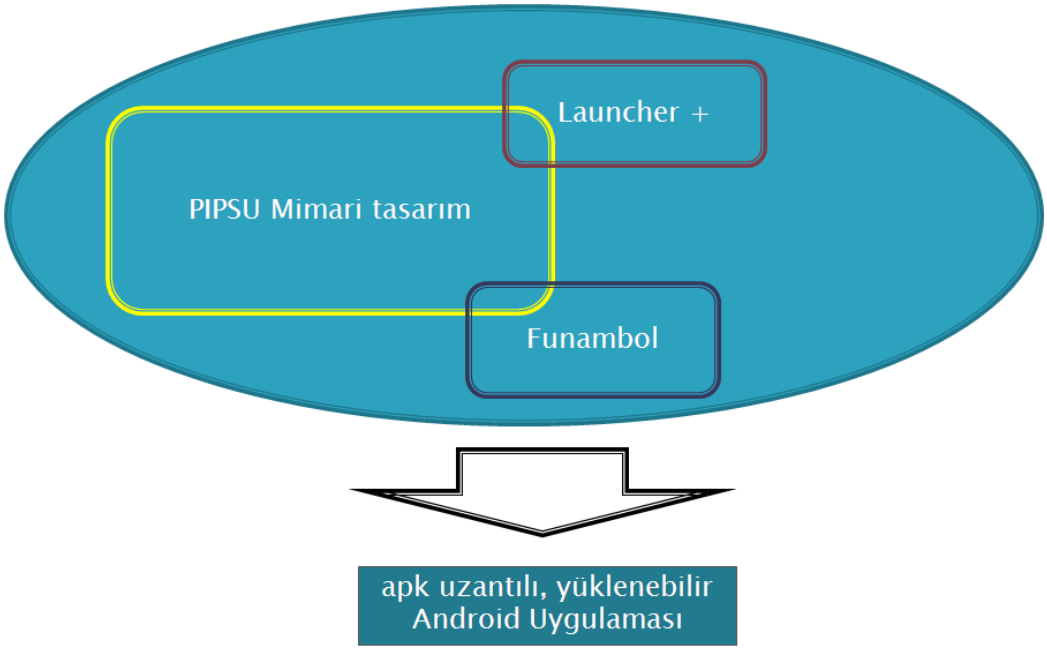
PIPSU-L : Temelde içsel olmayan bir Android uygulaması olarak tasarlanan bu bileşen; kullanıcı ara birimi olarak bir Android Launcher ve bu Launcher'a gerekli servisleri veren bir uygulama katmanından (Application Framework) oluşmaktadır.

PIPSU-C : PIPSU-L tabanlı istemcilere güvenilir ve devamlılığı olan, bulut tabanlı servisler vermeyi hedefleyen PIPSU platformu bileşenidir.



Şekil3.1PIPSU Platform ÜstDüzeyMimarisi

Şekil 3.1'da PIPSU platformunun iki temel bileşeni olan PIPSU-L ve PIPSU-C görülmektedir. Bunlardan PIPSU-L istemci tarafını oluşturan bileşen olup, mobil platformda kullanıcı etkileşiminden ve bunun sonucunda tetiklenen temel senkronizasyon ve veri güvenliğinin sağlanmasından sorumludur. İkincil sistem bileşeni olan PIPSU-C ise farklı mobil platformlar için geliştirilmiş PIPSU-L'lerin soyutlayarak PIPSU sunucusuna taşıdıkları kişisel verilerin depolanması ve korunmasından sorumludur.



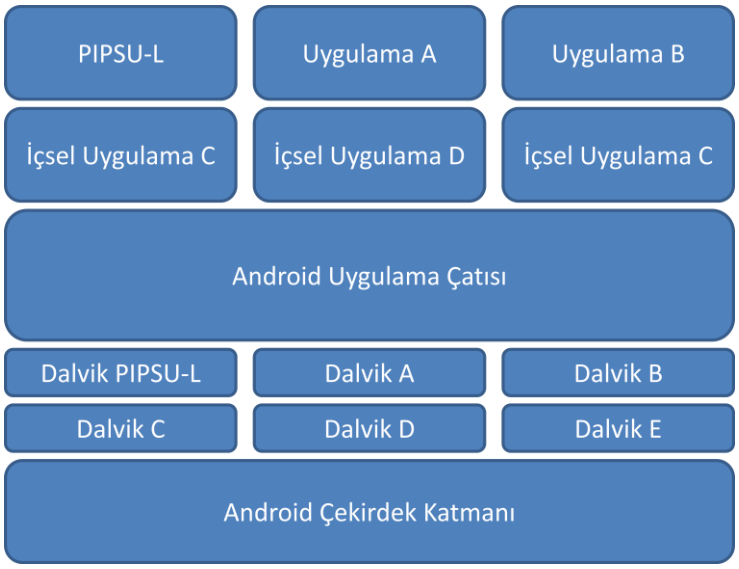
Şekil 3.2PIPSU geliştirilmesinde kullanılan açık kaynak projelerin mimarisel ilişkisi

PIPSU'nun mimari tasarımı herhangi bir projeden veya yapıdan bağımsız olarak ortaya konulmuştur. Önerilen mimarinin gerçeklenmesi için ise Launcher+ [22] ve Funambol [23] açık kaynak projelerinden faydalanılmıştır. Bahsi geçen projeler ile PIPSU platformunun ilişkisi Şekil 3.2'degösterilmiştir. Bu tip bir yaklaşım ile PIPSU'yu yazılımsal olarak daha hızlı ve az hatalı gerçeklemek mümkün olmuştur. Launcher+ projesi ile kullanıcının PIPSU işlevlerini arayüz üzerinden tetiklemesine olanak sağlanmıştır. Funambol projesi ile de kullanıcı verilerinin SyncML standardında sunucuya aktarılması mümkün olmuştur. Gerçeklenen PIPSU platformunda PIPSU-C işlevi Funambol sunucuları ile gerçeklenmiş ve test edilmiştir. PIPSU bir veri eşgüdümü standardı olan SyncML'i desteklemektedir, dolayısıyla başka SyncML destekli bulut servisleri ile de çalışması mümkündür.

3.2. PIPSU-L MODÜLER YAPISI VE SİSTEM MİMARİSİ

PIPSU, Android platformunun herhangi bir Android uygulamasına sunduğu olanakları(Android SDK ve Android Eclipse Plug-in) kullanılarak geliştirilmiştir. Mevcut Android tabanlı uygulamalardan temel farkı kullanıcı kişisel bilgilerine yönelik

güvenlik sağlamayı hedefleyen bir Android uygulaması olması ve internet tabanlı bir bulut servisinden oluşmasıdır.



Şekil3.3 PIPSU Platform Üst Düzey Mimarisi

Şekil 3.3'de PIPSU-L Android uygulamasının Android platformu üzerindeki yeri ve bu uygulamanın Dalvik sanal makinası üzerindeki izi gösterilmektedir. PIPSU-L, Android platformunda bulunan diğer içsel olamayan Android uygulamalarıyla aynı uygulama yapısına sahiptir. Bu bağlamda, PIPSU-L uygulaması, çalışma zamanı içerisinde sistem kullanıcısı tarafından kurulabilme, kaldırılabilme ve durdurulabilme haklarıyla kurulmaktadır.Bu sade yaklaşım PIPSU-L'nin Android tarafından standartlaştırılan API'lerin kullanılması yolu ile geliştirilmesini mümkün kılmaktadır ve mobil işletim sistemine ve versiyonuna özel bir ihtiyacasahipdeğildir. PIPSU-L uygulaması, Android mobil platformun kendisi (dolayısıyla mobil platform kullanıcısının kendisiyle) ile iki temel şekildeetkileşmektedir: (i) Kullanıcı tarafından oluşturulan veya başlatılan olaylar ve (ii) sistem kaynakları ve arabirimleri tarafından oluşturulan veya başlatılan olaylar.

Şekil3.4 PIPSU Platform Üst Düzey Mimarisi

Şekil 3.4 PIPSU platformunun sahip olduğu katmanlar ve bu katmanların kullanıcı kaynaklı veya platform kaynaklı olaylar ile tetiklenme akışını göstermektedir.

Kullanıcı temelli olaylar: Kullanıcının PIPSU-L launcher üzerinden bir kısayol yada PIPSU fonksiyon uygulamasını çağırması sonucu oluşur ve çağrısı yapılan fonksiyona göre şekillenir.

Sistem temelli olaylar: PIPSU-C servislerinin PIPSU-L'yi çalışma zamanında çağırmaları sonucu yada mobil işletim sistemininkendisi tarafından oluşur. Sistem temelli olaylar Kullanıcı temelli olaylarin tersi yönünde bir akışa sahiptirler.

3.2.1. PIPSU-L SANDBOX MODELİ

Linux çekirdek yapısının Android işletim sistemine sağlamış olduğu kullanıcı ve dosya sistemi erişim[29] yaklaşımından esinlenerek kendi uygulama sandbox modelini geliştirmeye çalışmıştır. Buna göre PIPSU-L Android uygulaması sistem üzerinde

kurulmaya başlandığında bir UID (user id) ile hayata başlamaktadır. Bu UID aracılığı ile diger Android uygulamalarının ve kullanıcısının ulaşamadığı bir dizin ve işlem (process) yapısı oluşturulmuş olunur. Böylece PIPSU-L, Android işletim sistemi üzerinde kullanacağı dizin yapısı ve erişim haklarını Android platformunun ortak erişim haklarının dışında PIPSU-L'ye özel bir yapıya taşımış olmaktadır. PIPSU-L Android platformunda sahip olduğu UID aracılığı ile geçici alt işlemleri ile birlikte tüm fonksiyonlarını ve verilerini Android platform kullanıcılarından bağımsız ve izole çalıştırabilmektedir.

3.3. PIPSU-L KATMANLI MİMARİSİ

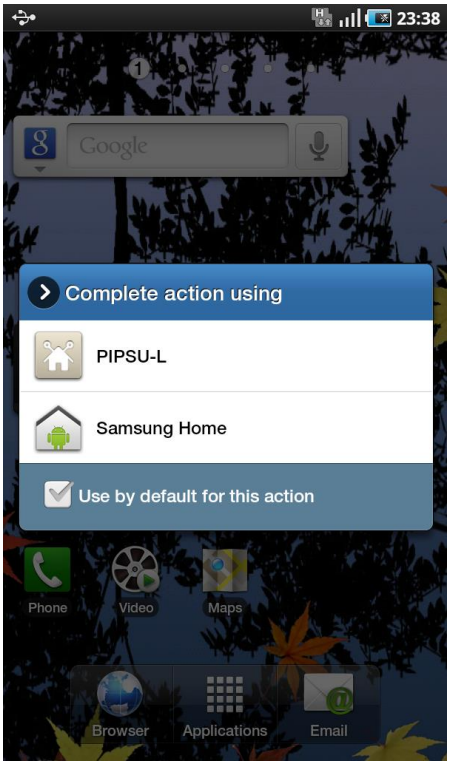
PIPSU-L kullanıcı verilerini sahip olduğu Launcher arabirimi ile diğer uygulamalardan ve sistemden izole fakat kullanıcı kullanıp kiplerini değiştirmeksizin sağlayabilen bir sunum katmanına sahiptir. Temelde bir launcher olarak geliştirilen PIPSU-L Sandbox'u içerisinde güvenli şekilde sakladığı kullanıcı kişisel verilerini yine aynı amaçla sadece kendi sahip olduğu kullanıcı arayüzleriyle başka uygulamaların erişimlerine izin vermeksizin kullandırmayı sağlayan bir kullanım arabirimine sahiptir.

PIPSU-L uygulama mimarisi dört ana katmandan oluşmaktadır. Bu katmanlar Şekil 3.5'de alt bileşenleri ile birlikte gösterilmiştir.

Şekil 3.2.1.5 PIPSU-L katmanlı yapısı ve alt bileşenleri

3.3.1. PIPSU-L SUNUM KATMANI

PIPSU-L çözümü alışılmış Android grafik uygulamalarının dışında, bir launcher uygulaması olarak gerçeklenmiştir. Sunum katmanında Launcher+ açık kaynak kod projesindenden yararlanılmıştır [22]. PIPSU-L, kullanıcıya uygulama kısayolu ile erişilebilen bir uygulamadan farklı olarak bir launcher arayüzü olarak tasarlanmıştır. Bu bağlamda PIPSU-L kullanıcı ara birimi olarak bir Android Launcher ve bu Launcher'a gerekli servisleri veren bir modüller katmanından oluşmaktadır.



Şekil 3.3.1.6 Launcherlar Arasında Sıralanmış PIPSU-L Uygulaması

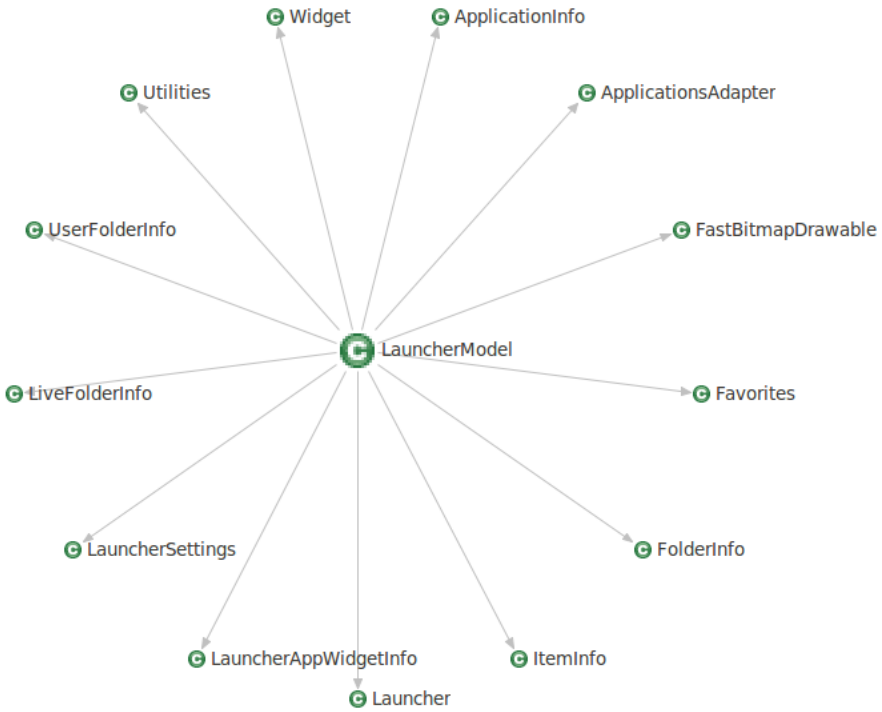
Şekil 3.6'da PIPSU Launcher uygulamasının, bir Android cihazında mevcut Launcherlar arasında sıralanması gösterilmektedir.

Şekil 3.7'de ise PIPSU-L Launcher arayüz bileşenleri ve PIPSU-L ayarlar kullanım arayüzlerine dair bileşenler gösterilmektedir.

Şekil 3.3.1.7 PIPSU-L Launcher Bileşenleri

Bu katman bileşenleri amaçlarına göre Widget'lar ve Activity'ler olarak iki ana bölüme ayrılmaktadır. Widget'lar Launcher üzerinde yapılacak işlemlerin farklı arayüzlere taşımaksızın aynı arayüz üzerinde yapılmasını sağlamaktadır. Activity'ler ise PIPSU-L uygulamasının ayarları ve kurulum arayüzlerini sağlamakta olup bunlar uygulamanın ilk

kurulumu aşamasında ihtiyaç duyulan ve sonrasında kullanıcınınkullandığı arayüzlerdirler.



Şekil 3.3.1.8 PIPSU-L Launcher Bileşenleri

Şekil 3.8'de PIPSU-L Launcher modülünün temel sınıflarıgösterilmiştir. Yıldız modeli temel alınarak gösterilen bu sınıf diyagramına göre PIPSU-L uygulamasının temel modelinin LauncherModel olupdiğer sunum katman temel sınıfları bu sınıftan temel yetkilerini alabilmektedirler. LauncherModel temel sınıfından türeyen PIPSU-L sınıfları temelde şu işlevleri gerçeklemektedirler:

- Launcher ayarlarının yönetimi ve sağlanması.
- Launcher üzerinde bulunacak Widget ve benzer görsel bileşenlerinin temel yetkileri ve organizasyonun sağlanması.
- Kullanıcı kişisel verilerine erişimi sağlayan temel uygulama kısayollarının sağlanması. Örneğin Mesajlar veya Telefon rehberine erişim.

Burada kullanıcı arabirimi PIPSU tarafından kullanıcının erişim yöntemleri ve fonksiyonalitesine göre iki gruba ayrılmaktadır.

PIPSU Widget'ları: PIPSU-L sunum katmanında kullanıcı kişisel bilgilerine yönelik yapılabilecek işlemleri ve bu işlemlerin son durumlarını gösterebilen bilgi amaclı yapılardır. PIPSU-L Widget bileşenleri Launcher+ adındaki açık kaynak kodlu projeden yararlanılarak tasarlanılmış olup mimarisel olarak kendi içerisinde üç ana katmanlıdır.

Şekil 3.3.1.9 PIPSU-L WidgetKatmanlı Mimari

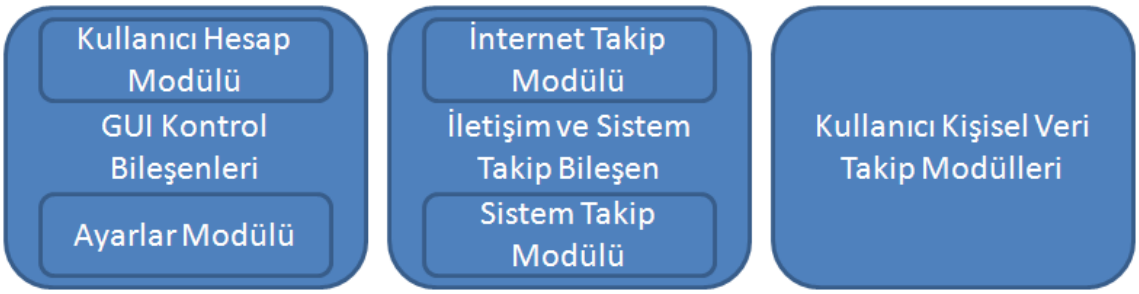
PIPSU-L, Launcher+ projesinin sahip olduğu launcher ve widget katmanlarından yararlanarak, bunların Android API aracılığı ile Webkit ile güvenli şekilde iletişimi ve kontrolünü sağlamıştır.

PIPSU Activity'leri: PIPSU platfrom ayarları ve kullanıcı veri senkronizasyon ayarlarını yönettiği kullanıcı arabirimlerini activity'ler halinde geliştirerek, bu arayüzlerin çalışma zamanı hayat döngülerini kolayca yönetebilmektedir.

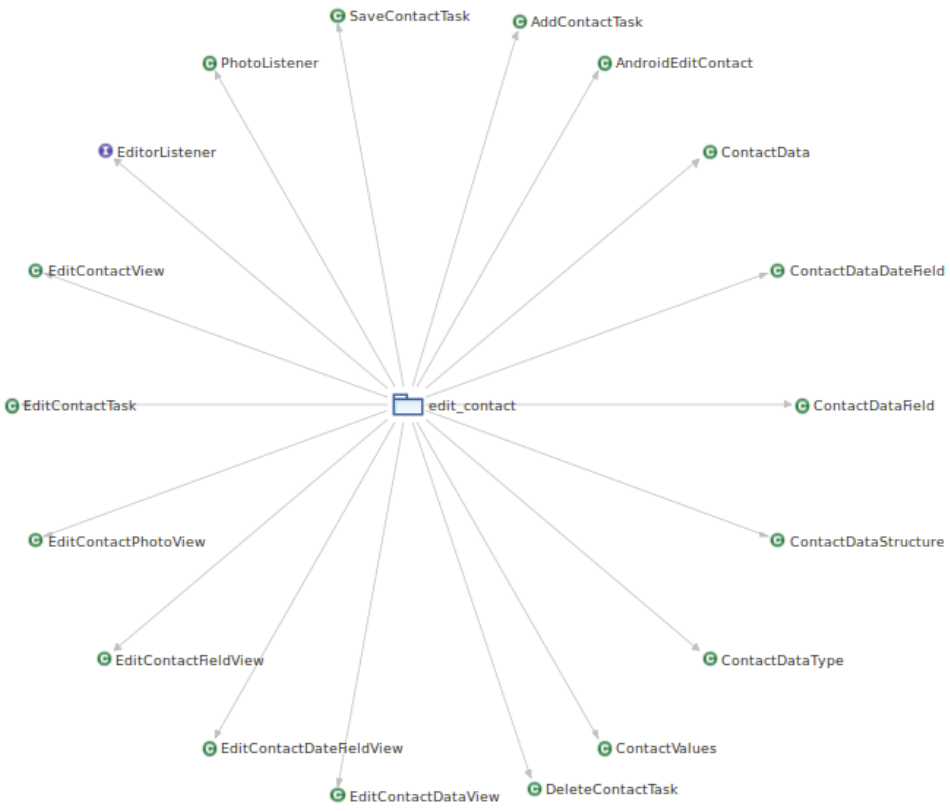
3.3.2. PIPSU-L KONTROL KATMANI

PIPSU-L mimarisindeki ikinci katman kontrol katmanıdır. Bu katman, PIPSU-L'nin çalıştığı platformda bulunan kullanıcı kişisel bilgilerinin, PIPSU tarafından sağlanan güvenli depolama alanlarına taşınmasını sağlamaktadır. Aynı zamanda, ilgili platformda bulunan kullanıcı kişisel bilgilerinin PIPSU tarafından anlamlandırılması görevini üstlenmektedir. Kullanıcı kişisel bilgileri, ilgili platform tarafından güncellenediğinde veya yeni bilgiler eklendiğinde, bu katman aracılığı ile ilgili değişiklik takip edilir ve depolama katmanına iletilir. PIPSU-L'nin işlevsel olarak sahip olduğu güvenlik katmanı

Linux çekirdeğinde bulunan ve her Android uygulamasının yaralanabildiği *Application Sandbox* modelinedayandırılmıştır. PIPSU-L'nin depolama alanına erişimi ve bu alanda saklanan verilerin izinsiz erişime kapatılması, bu katman ile amaca uygun şekilde yapılmaktadır. Kontrol katmanı tasarımında kullanılan modüller şekil 3.10'da gösterilmiştir. Bu katmanda kullanıcı hesap kontrolü ve PIPSU-Lye dair ayarların organize edildiği modüller vardır. Bu modüller, internet bağlantı kontrolü gibi olay tabanlive peryodik denetim altinda tutulması gereken işleri yapmaktadır.



Şekil 3.3.2.10 PIPSU-L Kontrol Katman Bileşenleri



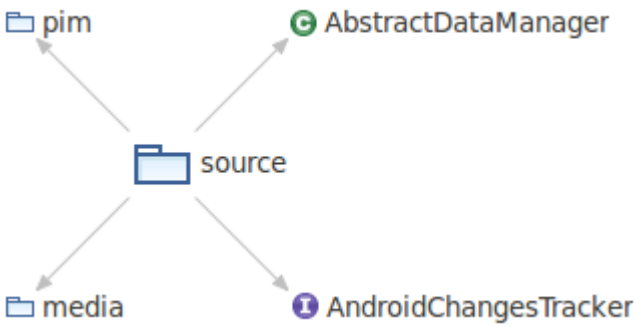
Şekil 3.3.2.11Kullanıcı Hesap Bileşenleri

Yukarıda Şekil 3.10 ve Şekil 3.11 kullanıcı hesap modülüne dair bileşenleri göstermektedir. EditorListener arayüzü ile kullanıcı hesabına dair değişiklikler veya yeni kullanıcı yaratılması sağlanmaktadır.

3.3.3. PIPSU-L SOYUTLAMA KATMANI

PIPSU-L mimarisinde üçüncü katman soyutlama katmanı olarak ortaya çıkmaktadır. Bu katman, kullanıcı kişisel bilgilerinin platform bağımsız şekilde soyutlandırılarak anlamlılandırıldığı katmandır. Bundan dolayı kullanıcı kişisel bilgileri PIPSU-C üzerinde platform bağımsız olarak saklanabilmesi hedeflenmiştir. Bu soyutlama, platform düzeyinde PIPSU-L temel veri türleri ile sağlanmaktadır. Fakat PIPSU-C tarafına taşınması ve burada anlamlılandırılması için SyncML format yapısına ihtiyaç duyulmuştur. Yani PIPSU-L tarafından platform bağımsız veri türlerine dönüştürülen kullanıcı kişisel bilgileri, SyncML mesaj yapıları kullanılarak; kullanıcı kişisel bilgilerinin PIPSU-C üzerinde platfrom bağımsız depolanması sağılanmıştır.

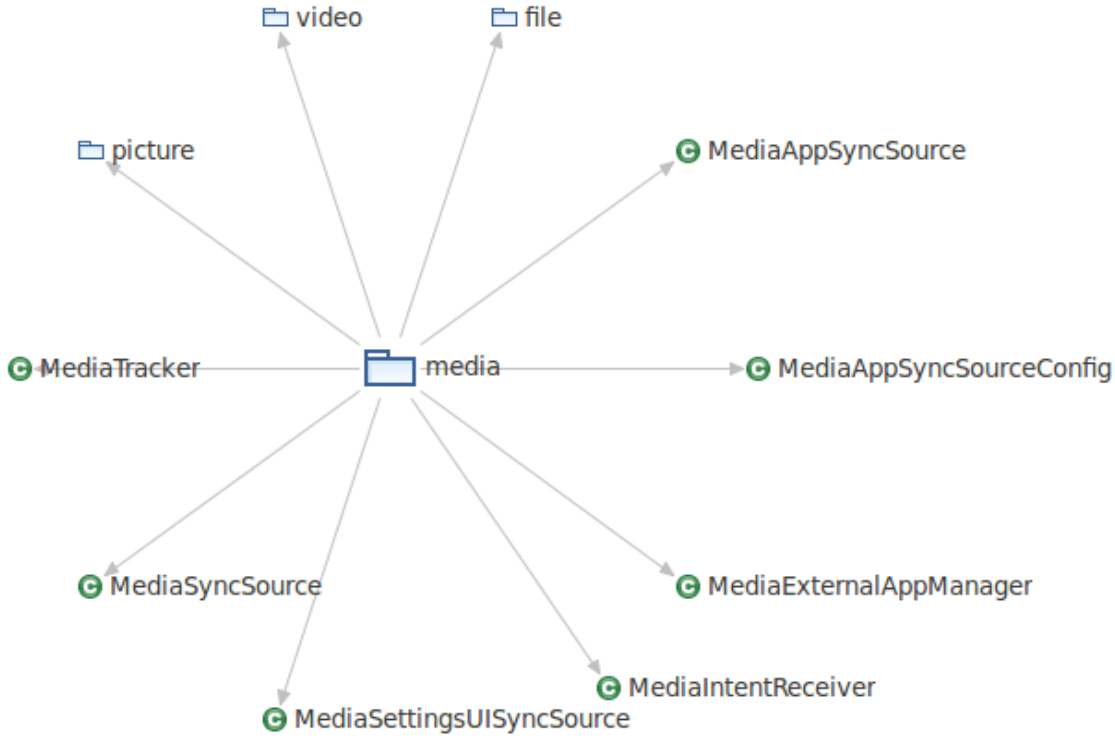
Şekil 3.3.2.12 Soyutlama Katmanı Bileşenleri



Şekil 3.3.2.13 Kullanıcı Kişisel Bilgi Soyutlama Paket Bileşenleri

Kullanıcı kişsel veri türleri PIPSU platform tarafından iki ana parametre ile saklanmaktadır. Birincil parameter kişisel verilerin ilgili platform içerisinde uygun soyut tipte saklanmasını sağlamaktır. İkincil parameter ise çalışma zamanı içerisinde bu verinin değişimlerini gözlemlemekte ve uygun durumlarda PIPSU-L üzerindeki değişimleri PIPSU-C ye taşımakta kullanılmaktadır. Kullanıcı kişisel bilgileri temelde 2 ana sınıfa ayrılmıştır.

- Medaya türündeki kullanıcı kişisel bilgileri: Resim, Müzik ve Video kayıtları
- PIM(Personal Information Management) : Kontaklar, Mesajlar, Takvim kayıtları vb.

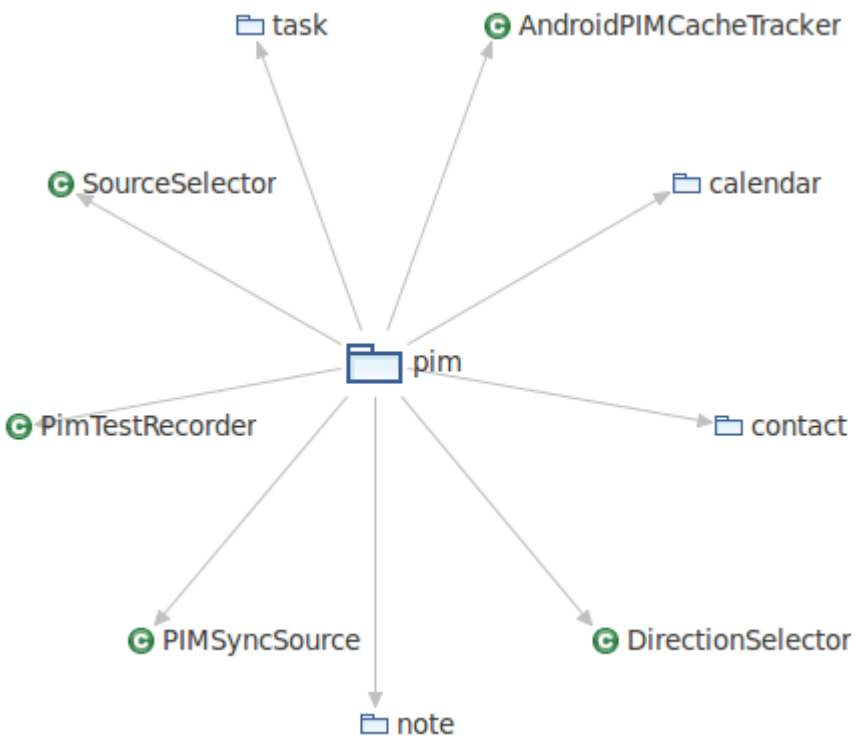


Şekil 3.3.2.11 Medya Türü Soyutlama Bileşenleri

Medya türündeki veriler ;

- Türden bağımsız ikilik veri (binary) türündeki veriler
- Resim türündeki veriler
- Video türündeki veriler

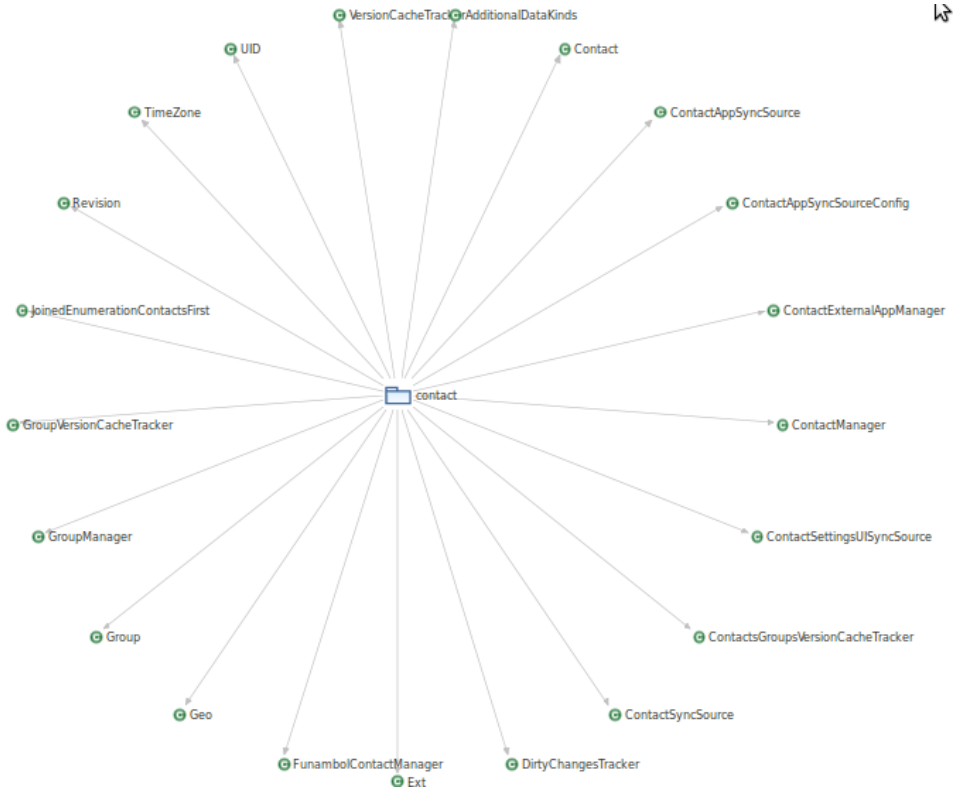
olarak üç türde saklanmaktadırlar. Bu tiplerin soyutlanması MediaSyncSource veri tipi ve bu tipten türeyen yeni tipler aracılığı ile sağlanmaktadır. Bu tiplerin soyutlanmaları ve yorumlanmalarına dair konfigurasyonları MediaAppSyncSourceConfig tipi ve bu tipten türeyen yeni tipler ile organize edlimektedir. MediaTracker tipi ve bu tipten türeyen yeni tipler ile medya türünde soyutlanmış verilerin değişimleri gözlenmektedir.MediaIntentReceiver sınıfı ise bu türdeki kullanıcı kişisel verilerinin değişimlerini bir üst katmana taşımakla görevli olay türüdür.



Şekil 3.3.2.15 PIM Soyutlama Bileşenleri

PIPSU-L uygulaması peryodik olarak yapması gereken işleri task türündenki temel sınıf aracılığı ile bu yeteneği geliştirilen diğer sınıflara taşımaktadır. Böylece PIPSU-L kullanıcı kişisel bilgilerinin local değişimlerini PIPSU-C üzerine peryodik olarak taşıyabilmektedirler. DirectionSelector temel sınıfı ise senkronizasyon işleminin hangi

yönde yapılacağını organize etmektedir. PIPSU-L DirectionSelector temel sınıfından türeyen modüller değişikliğin mobil platformda olması veya PIPSU-C üzerinde (Örneğin: aynı hesaba sahip başka bir mobil cihazda bulunan PIPSU-L uygulaması tarafından değiştirilen kullanıcı kişisel verisi.) olaması durumuna göre senkronizayon gönünü belirlemektedir.



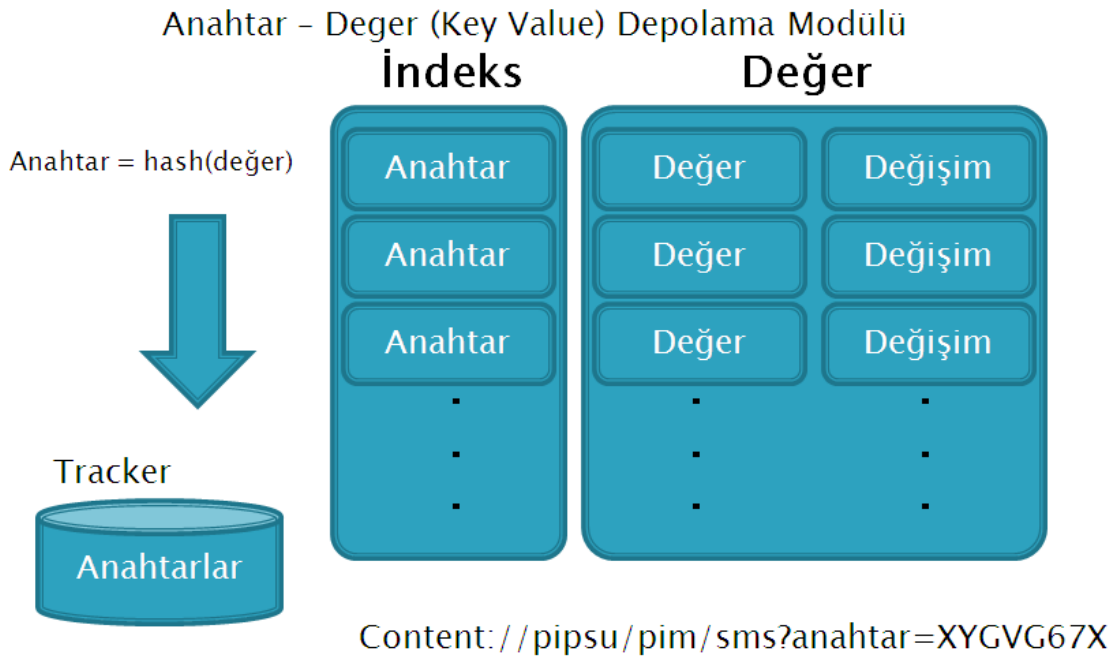
Şekil 3.3.2.16 PIM Kontak Türü Implementsayon Bileşenleri

PIPSU-L Senkronizasyonu sağlanacak kullanıcı kişisel bilgilerini değişikliği; zaman dilimi (Timezone) , revision farkı ve ilgili verinin hash algoristması çıkıtısının sağladığı farklar gözönünde bulundurularak değişikliğin olup olmadığına karar vermektedir.

3.3.4. PIPSU-L DEPOLAMA KATMANI

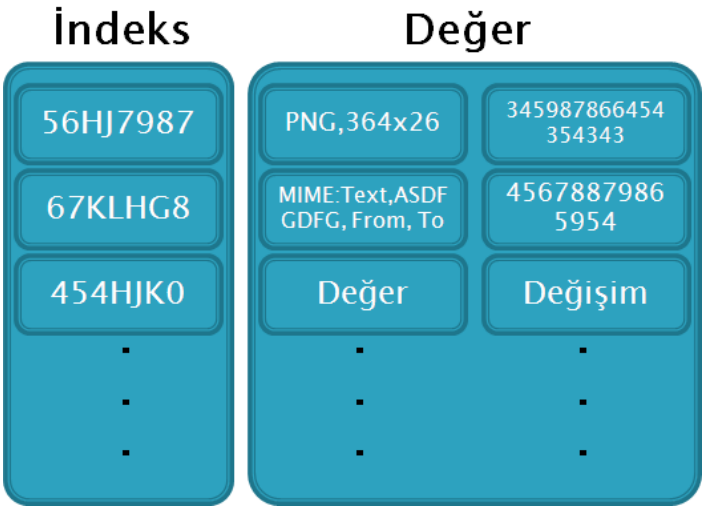
Dördüncü katman depolama katmanı olup; kendi içinde iki ana bölüme ayrılmaktadır. İlk bölüm kullanıcı kişisel bilgilerinin ilgili platform içersinde güvenli şekilde saklanmasını hedefleyen bölümdür. Bu bölüm Android platformunun application sandbox modelinden esinlenilerek tasarlanılmıştır. Temelde Key-Value NoSQL türünde

bir veritabanına benzeyen bu katmanın genel amacı, kullanıcı kişsel bilgilerini ilgili platform içersinde güvenli şekilde barındırabilmesidir. Bu yapı "bir kez yaz çok kez oku" (write once read many) yaklaşımından esinlenilerek; temel amaçları yaratıldıktan sonra sıkça kullanılması gereken kullanıcı kişsel verilerine, hızlı ve güvenli erişimi sağlamaktır.



Şekil 3.3.2.17 PIPSU-L Çevrim Dışı Kip Kullanıcı Veri Deposu

PIPSU çevrim dışı kip deposu iki ana bölümden oluşmaktadır. Bunlar Indeks ve Değer modülleri olaraka adlandırılmaktadır.

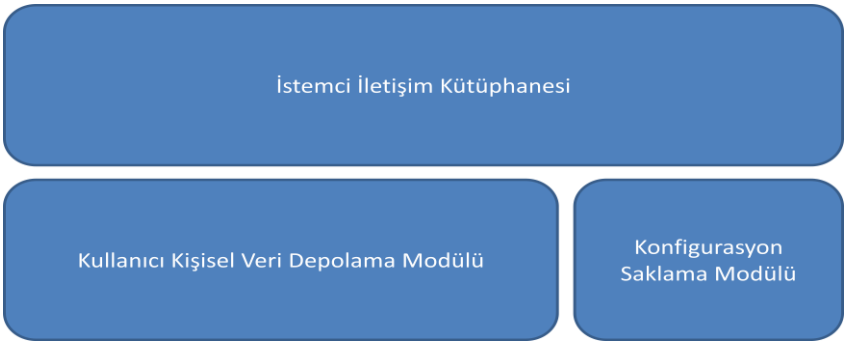


Şekil 3.3.2.18 PIPSU-L Örnek Kullanıcı Kişisel Verisinin Depolanması

Indeks modülü depoda bulunan kullanıcı kişisel bilgilerine hızlı erişim için tasarlanmış olup her index kendi içerisinde temsil ettiği değere dair bilgiyide barındırmaktadır. Bir Indeks değeri tutulan kullanıcı kişisel verisinin MD5 Hash algoritması ile üretilen sonuca eşdeğerdir.

Değer modülü ise kullanıcı kişisel verilerinin saklandığı ve bu verilerin değişimlerinin tutulmasından sorumlu modüldür. Bu modül aracılığı ile veriler önce dahili bellekte sonrada bellekteki izleri ise kalıcı diske yazılmaktadır. Bir kişisel bilginin değişimi ise zaman bilgisi (timezone + timestamp) ve enson hash bilgisinin toplamı olarak ele alınmaktadır.

İkinci bölüm ise bulut tabanlı bir depolama alanıdır. Kullanım amacı; platformdan bağımsız kullanıcı kişisel bilgilerinin farklı platformlara taşınabilmesini sağlamaktır. Bu katmanın implementasyonuda, kullanıcı kişisel bilgilerini standart formatlarda saklamak amacıyla SyncML ‘in sağladığı yaklaşımdan ve standartlardan yararlanılmıştır. Implementasyonda ise SyncML implementasyonlarından olan açık kaynak kodlu Funambol[23] dan aktif olarak yararlanılmıştır.



Şekil 3.3.2.19 Depolama Katman Bileşenleri

3.4. TEMEL PIPSU-L KULLANIM SENARYOSU

Temelde bir Android uygulaması olan PIPSU-L, Android uygulama paketi (APK[24]) olarak Android tabanlı cihazlara 2 şekilde kurulabilir.

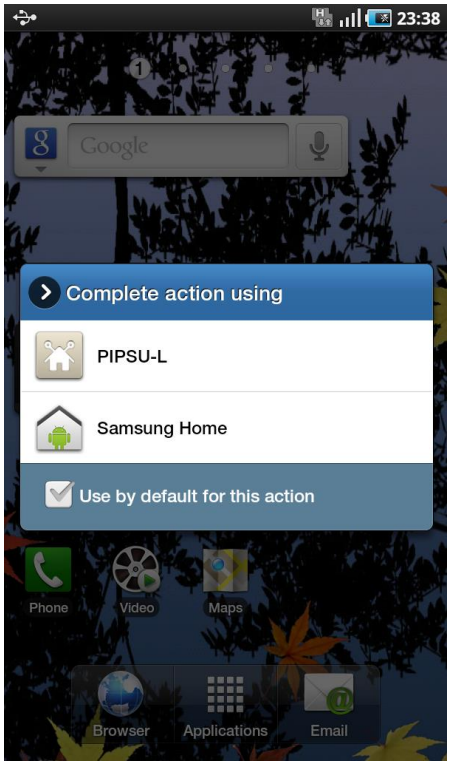
- Bir Masa üstü veya Laptop bilgisayardan PIPSU Android uygulamasının ilgili Android cihazına, cihaz üzerine bir sdcard ile taşınması ve cihazdan bu sdcard'a erişilip PIPSU uygulamasının kurulması.
- Yine bir Laptop veya Desktop cihaz ile USB ara birimi aracılığı ile bağlanılmış bir Android tabanlı cihazda hata ayıklama fonksiyonunun açılmasından sonra yine PIPSU uygulamasının ADB(Android Debug Bridge[25]) komutları aracılığı ile kurulması.

Android tabanlı cihazda kurulmak istenen PIPSU-L uygulaması, her Android uygulamasının temelde yapmak zorunda oluğu sistem kaynaklarına erişim hakklarının PIPSU-L'ye cihaz kullanıcısı tarafından verilmesini gerektirmektedir.

PIPSU-L Uygulaması Cihaz kullanıcısına sistem öz kaynakları olan;

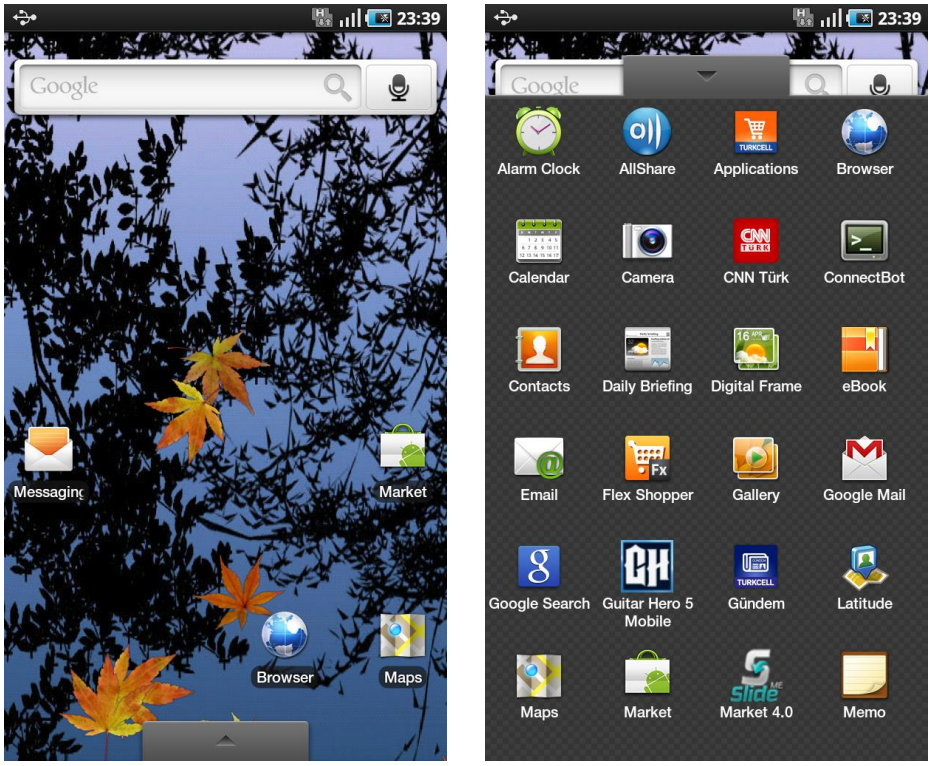
- İnternet kullanımı,
- Sdcard erişimi,
- İletişim(Contacts) verilerine erişim,
- Mesajlaşma verilerine erişim,
- Multi media verilerine erişim,
- Sim Kart erişimi

kullanımına dair uyarılarında bulunur ve bu erişimlerin onaylanmasını ister. İlgili kullanıcı onayını edinen PIPSU-L uygulaması kurulmuş olur.



Şekil 3.3.2.20 Kurulu Launcher listesinde PIPSU-L uygulaması

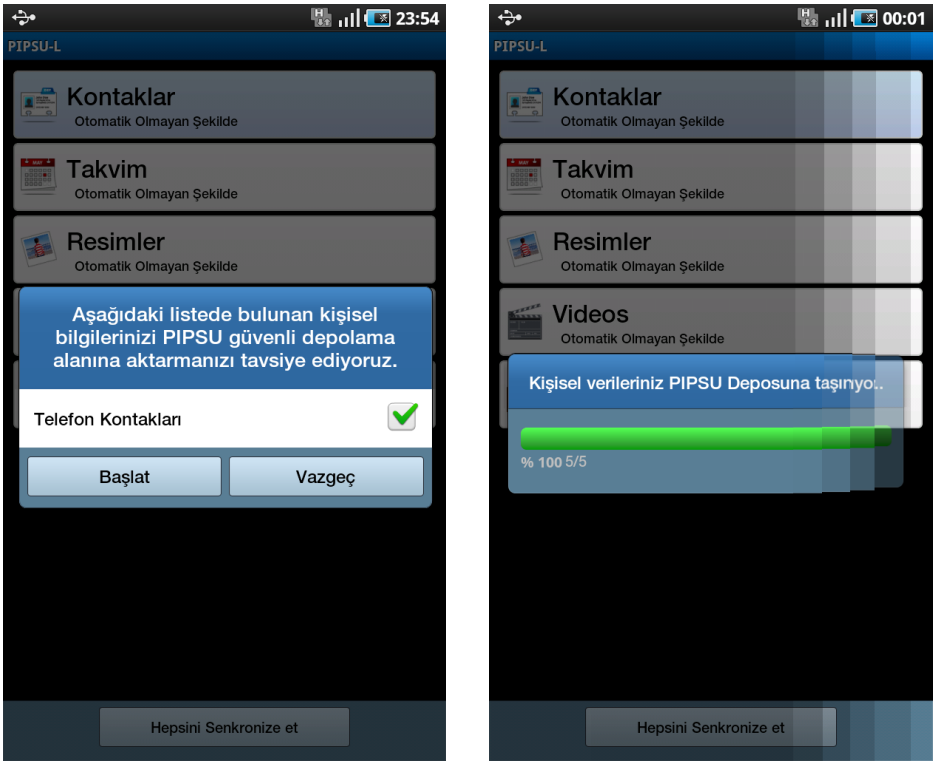
Android Launcher uygulaması olarak geliştirilen PIPSU-L, cihaz kullanıcısının cihaz üzerindeki home butonuna basması durumunda, Android işletim sistemi tarafından varsayılan launcher(Android işletim sisteminde içsel Android uygulaması olarak bulunan Launcher) ile PIPSU arasında bir seçim yapılması istenecektir. Yani çalışma zamanında Android tabanlı cihazın sahip olduğu home butonunun ürettiği eventi yakalayabilme yeteneğine sahip bütün launcher tabanlı uygulamalar, bu eventin her üretilmesinde Android işletim sistemice sıralanacak olup kullanıcı tarafından kullanılmak istenilen uygulamanın seçilmesi gerekecektir. PIPSU launcher tabanlı Android uygulamasının kullanıcı tarafından bir kez varsayılan launcher olarak tanımlanması bu belirsizliği ortadan kaldıracaktır. PIPSU varsayılan laucher olarak tanımlandıktan sonra artık Android tabanlı cihazın ürettiği home evenleri PIPSU tarafından yakalanabilecek ve PIPSU aktif uygulama kipine geçebilecektir.



(a) (b)

Şekil 3.3.2.21 PIPSU-L uygulaması (a) Launcher kısayolu (b) Uygulamalar kisayolları

PIPSU-L ilk olarak cihazda sim kart olup olmadığını eğer var ise buradaki iletişim verilerini yedeklemek ister.



(a) (b)

Şekil 3.3.2.22 PIPSU-L KullanıcıKişiselVeriAktarımİşlemi (a) Sorgulaması (b) İşlem Durum Göstergesi

PIPSU-L, sim kart'ta bulunan iletişim verileri ile birlikte dosya sisteminde bulunan multi media dosyalarını, mesajları ve notları da yedekler.

Şekil 3.3.2.23 PIPSU-L Veri Aktarma İşleminin Ertelenmesi

Temelde bu işlem bir veri import işlemine eşdeğerdir. Bu yedekleme işlemi iletişim verileri için PIPSU-L'in iki farklı çalışma kipi için kullanımı farklılaşmaktadır.

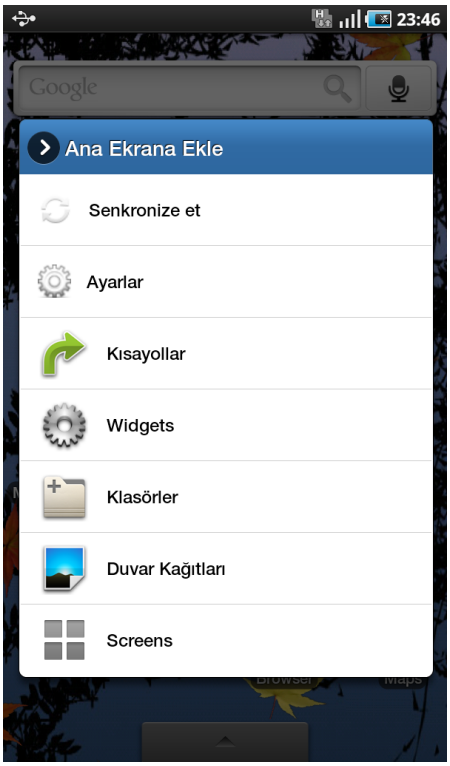
- PIPSU-L çevrim içi çalışma kipi: Bu kipte PIPSU-L yedeklemeyi hedeflediği veri türlerini, iletişim verileri ve diğer veriler olmak üzere PIPSU-C üzerinde yedekler. PIPSU-L'nin çevrim içi ilk çalışma kipinde, kullanıcı veri erişimi ,yani daha önce PIPSU-L aracılığı ile PIPSU-C ye yedeklenen ve şuanda PIPSU-L de bulunmayan iletişim verileri yada PIPSU-L aracılığı ile ilk defa iletişim verisi kaydı girildiğinde, PIPSU-L bu iletişim verisini PIPSU-C üzerinden okur ve PIPSU-L çevrim dışı çalışma kipi erişimi için sahip olduğu güvenli alanda saklar.

- PIPSU-L çevrim dışı çalışma kipi: PIPSU-L bu kipte sadece iletişim verilerini PIPSU-L depolama katmanında güvenli şekilde yedeklemeyi sağlar. Fakat PIPSU-L bu kipte PIPSU-L depolama katmanı ve PIPSU-C üzerinde yedeklediği bütün verilerin son durumlarını, yani enson yedekleme tarihine ve durumuna bağlı olarak bir değişiklik olup olmadığı durumunu PIPSU-L çevrim

dışı ve PIPSU-L çevrim içi modlarında takib eder ve PIPSU-L çevrim içi kipte bu bilgiye bağlı olarak değişikliği gözlenen kullanıcı verileri güvenli şekilde saklanmak amacıyla PIPSU-C üzerinde yedeklenir.

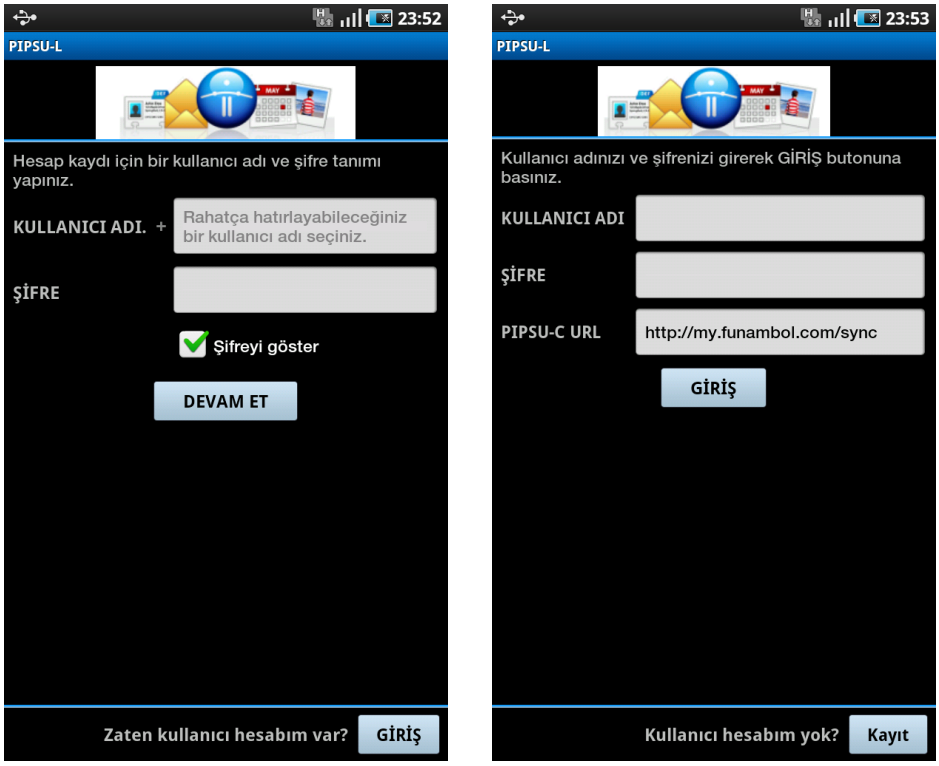
3.5. PIPSU PALTFORMUNDA PIPSU KULLANICI HESAP ROLÜ

PIPSU uygulaması kurulum sonrası kullanımına başlanabilmesi için; PIPSU uygulaması ilk açıldığı anda hesap tanımının yapılması gerekmektedir.



Şekil 3.3.2.24 PIPSU-L Kısayolları

PIPSU-L kısayolları listesi yukarıdaki şekilde gözükmektedir. Bu kısayollardan PIPSU senkronizasyon işlemiyle alakalı olan herhangi bir işlem seçilmesi durumunda PIPSU-L hesap tanımlama ekranına geçilerek kullanıcının hesap tanımlaması istenecektir.



(a) (b)

Şekil 3.3.2.25 PIPSU-L (a) Yeni Hesap Tanımlama Ekranı (b) Hesap ile Giriş Yapmak

Kullanıcı PIPSU-L uygulamasını kullanabilmek için yeni hesap tanımı yapabilir yada daha önce yarattığı başka bir hesabı kullanbilir.

PIPSU hesap tanımlanması kullanıcının kendisinin belirleyeceği bir kullanıcı adı ve şifreden oluşmaktadır.Bu hesap bilgileri sadece PIPSU tarfından erişilebilen bir alanda güvenli şekilde saklanmaktadır.Kullanıcının PIPSU üzerinde tanımladığı bu hesap PIPSU üzerinde aşağıda listesi verilen PIPSU fonksiyonlarının kullanımı için gereklidir.

- Yeni kontak listesi import'u
- Mevcut bir kontact'ın silinmesi
- Yeni bir kontact eklenmesi

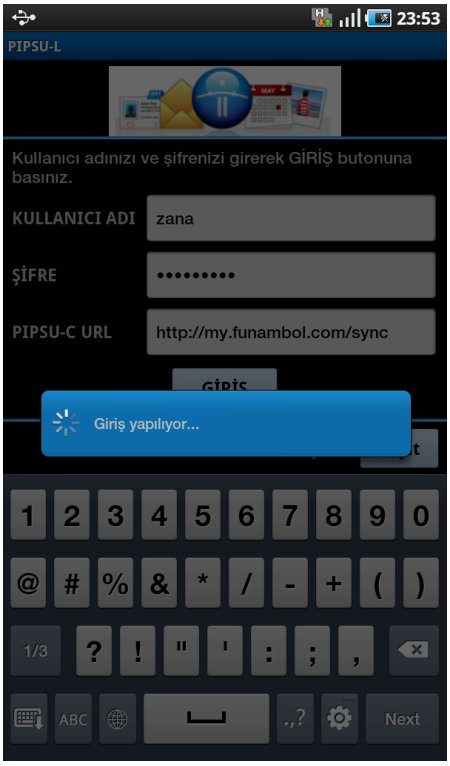
Sırasında gerekmektedir.

PIPSU üzerinde tanımlı kullanıcı hesaplarının birleşiminden elde ettiği bir kimlik bilgisi ile verdiği servisleri yönetir. İlgili kimlik bilgisi PIPSU düzeyinde tekil olup,

- PIPSU kullanıcı hesabı

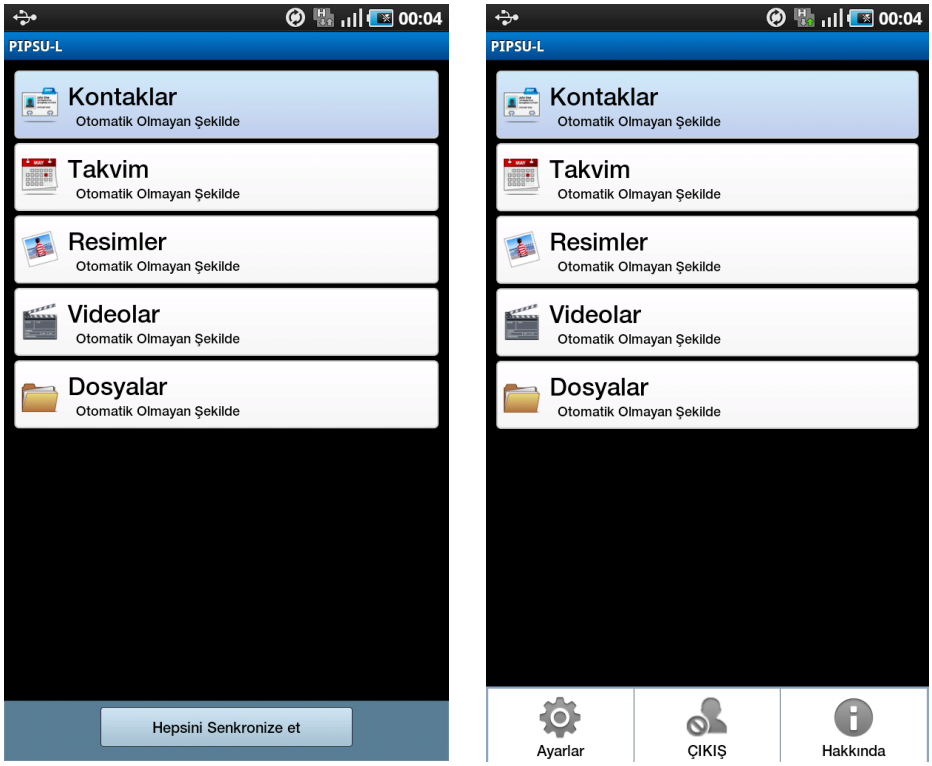
- Mobil cihazın IMEI kimliği
- Mobil cihaz üzerindeki işletim sistem bilgileri (Android işletim sistemi API'si ile edinilen fingerprint)

parametrelerinin birleşiminden oluşmaktadır.



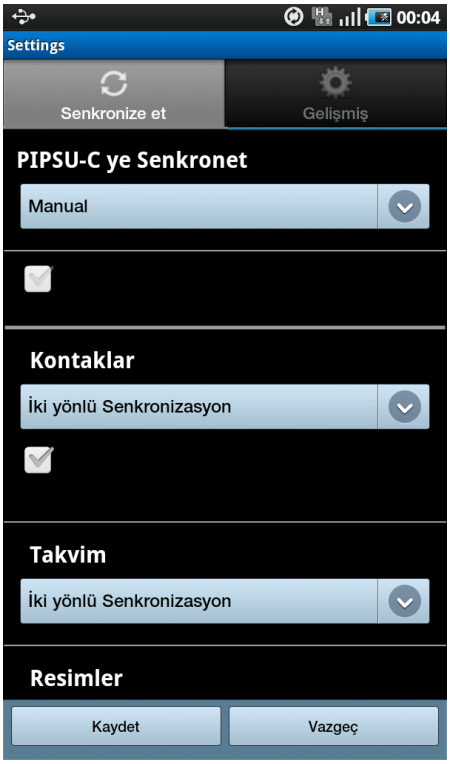
Şekil 3.3.2.26 PIPSU-L Giriş Animasyonu

PIPSU-L Kullanıcısı herhangi bir zaman diliminde bütün kişisel bilgilerini PIPSU-C ile senkronize edebilir.



(a) (b)

Şekil 3.3.2.27 PIPSU-L (a) Veri Senkronizasyon Arayüzü (b) Alt Menüleri



Şekil 3.3.2.28 SenkronizasyonAyarları

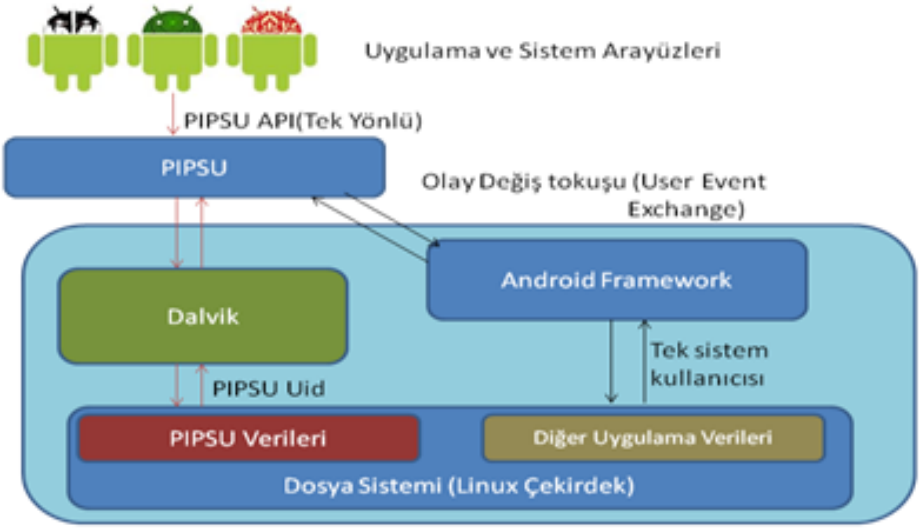
Ayrıca PIPSU platform kullanıcısının kişisel veri senkronizasyonun otomatize edilmesi de mümkündür. PIPSU üzerindeki senkronizasyon işlemi 2 alternatifle yapılabilmektedir.

- 2 Yönlü senkronizasyon : Bu model seçildiğinde PIPSU-L üzerindeki her değişim PIPSU-C ye aktarılmaktadır. Aynı şekilde PIPSU-C üzerindeki her değişim PIPSU-L ye de aktarilmaktadır. Burada aktarımın yönünü en son değişikliğin yapıldığı ortam baz alınması belirler.
- Tek yönlü aktarım : Bu modelde aktarımın yönü daima PIPSU-C ye doğrudur. Yani PIPSU-L deki değişimler PIPSU-C ye taşınır.

3.6. PIPSU VERİ GÜVENLİK MODELİ

PIPSU, alışılmışa gelmişgüvenlik modeller,inden farklı bir güvenlik modeline sahiptir. Sıradan bir Android uygulamasına sunulan kaynaklardan yararlanarak güvenlik modelini sağlayabilmektedir. PIPSU'nun sahipolduğu güvenlik modelini iki temel etmen altında incelenebilir:

PIPSU ya özel UID(Process Unique Identity): Bir çok Android uygulama geliştiricisi Android mobil platformuna yönelik geliştirdikleri uygulamalara özel UID tanımı yapmayıp, bunun sistem tarafından belirlenen varsayılan değerlerce atanmasına izinvermektedirler. Mobil platform varsayılan değerlerininsistemde yüklenmiş tüm mobil uygulamalarca bilinmesi birçok güvenlik açığını da beraberinde getirmektedir. Buna göre uygulamalar varsayılan degerler aracılığı ile birbirlerinin uygulama kaynaklarına(Application context) ve erişim haklarına sahipolabilmektedirler. PIPSU-L sahipolduğu bütün işlevlerini(process) kendi tanımladığı UID tanımları ile Linux işlev yönetim yaklaşımından esinlenerek sadece kendi yönetimi altına almış olup, diğer uygulamaların ve mobil platform varsayılan tanımlarının dışına taşımış olur.

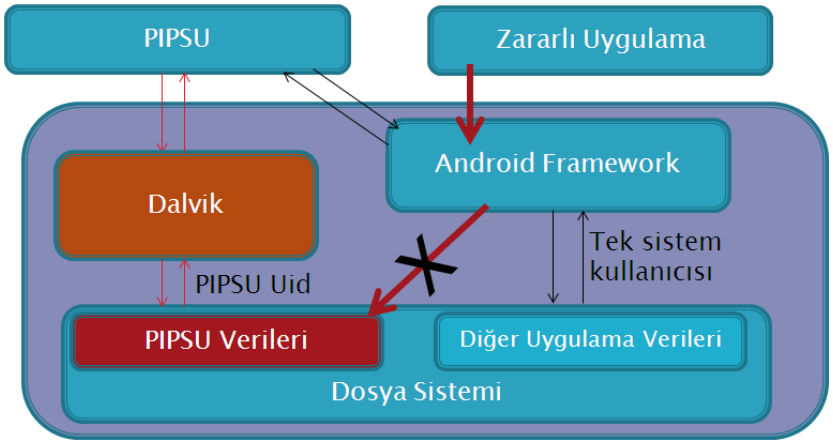


Şekil 3.3.2.29 PIPSU-L Uygulama – Veri Güvenlik Modeli

Şekil 3.29 PIPSU API'sinin mobil platform uygulamaları tarafından kullanımını göstermektedir. PIPSU, API arabirimi aracılığı ile kullanıcı kişsel verilerini, aynı platformda bulunan kötü amaclı uygulamalara karşın güvenliği sağlıyabilmektedir. Ayrıca PIPSU kişisel veri servisleri PIPSU API arabirimi aracılığı ile salt okunur olarak güvenle kullanılabilmesi sağlanabilmektedir. Burada PIPSU uygulamalasının sıradan bir Android uygulamasından temel farkı kendine özel UID ile hayata başlaması olup, böylece sadece PIPSU uygulaması tarafında kullanımı sağlanan diğer uygulamaların erişiminin mümkün olmadığı bir güvenli bölge sağlamıştır.

PIPSU salt okunur veri kullanım modeli: PIPSU sahip olduğu kendisine özel uygulama konteksi aracılığı ile ilgilimobil platform ile salt okunur bir etkileşim içerisindedir. Bu bağlamda, PIPSU,mobil platform tarafından sağlanan API'lerce değişen (veya ilk defa yaratılan) kullanıcı verilerini takipetmek(change tracking) için gene bu API'lerin kendisini kullanmaktadır. Mobil platform ile veri paylaşımı yine sadece çalışma zamanı içerisinde ilgili mobil platformun kullanıcı kişisel verilerine yönelik sağladığı fonksiyonaliteyi salt okunur veri paylaşımı ile kullanabilmektedir.Fakat kişisel verilerin saklandığı dosyas ve veri bölümü sadece PIPSU uygulaması tarafından erişilebilen bir bölüm olup diğer uygulamalarca erişilmemesi PIPSU UID tarafından garati altına alınmıştır.

Şekil 3.3.2.25'dePIPSU plafromununun ilgili mobil platformda bulunan uygulama ve API lerin salt okunur etkileşimi göstermektedir.



Şekil 3.3.2.30PIPSU-L Uygulama – Veri Güvenlik Modeli

Şekil 3.30'da mobil Android platformunda bulunan PIPSU Android uygulamasının zararlı uygulamalar tarafından etkilenmeyişigösterilmektedir. PIPSU veri güvenlik modeli ile PIPSU uygulama verileri ve dolayısıyla kullanıcı kişisel verilerine diğer Android uygulamaları erişememektedirler. Ayrıca,Android gibi mobil platformların uygulamalara sağladıkları framework API'leri sistem kaynakları ve kullanıcı kişisel verilerine dair bir çok açığıda beraberinde getirmektedirler. Android Framework gömülü veritabanı (embedded SQLite), veri erişim arabirimleri(Content Providers, Content Resolvers) ve uygulama içi yada uygulamalar arası haberleşme yöntemleri ile birlikte, Android uygulamaları tarafından kullanılmak üzere bir çok özellik sunmaktadırlar. PIPSU diğer Android uygulamalarından farklı olarak Android framework'ünü salt okunur kullanarak kullanıcı kişel verilerini kendi dosya sisteminde ve erişim hakları ile saklamaktadır.

4. TARTIŞMA VE SONUÇ

Günümüz mobil platformlarında bir çok güvenlik ve kullanım sorunu bulunmaktadır. Bu tez çalışmasında günümüz mobil platformlarda bulunan kullanıcı kişisel verilerine yönelik güvenlik açıklarını detaylı şekilde incelenmiş ve problemin kaynakları ortaya konulmuştur. Mobil platformların mevcut güvenlik açıkları ve bu konuda yapılan akademik ve ticari çalışmalar detaylı olarak incelenmiş, günümüz mobil güvenlik sorunu ile ilgili büyük resim ortaya konmuştur. Yapılan sorun analizleri ve mobil platformların mimarisel incelemeleri sonucu, Android mobil platformunu örneği bu tez kapsamında mimarisel ve implementasyon açısından incelenmiştir. Android mobil platfromu hedef alınarak gerçekleştirimini sağladığımız çözüm önerimiz olan PIPSU'nun işlevsel ve mimarisel tasarımı ortaya konulmuştur. PIPSU tasarımı ayrıca Funambol ve Launcher+ açık kaynak projeleri de kullanılarak gerçeklenmiştir. Tezin ana katkısı olan PIPSU tasarımının özellikleri:

- Kullanıcı kişisel verileri için güvenlik çözümü sağlamaktadır.
- Mobil platformdan bağımsız bir çözüm sunar.
- Bulut bilişim uyumludur (Cloud Computing Compatible).
- Sunum, servis ve depolama katmanlarını bir arada barındıran bir çözümdür
- SyncML protokol standardı ile farklı bulut sistemleriyle çalışabilmektedir.
- Aynı hesap altında çoklu cihaz desteği verebilmektedir.
- Dahili depolama fonksiyonu ile çevrimdışı(Bulut servislerine bağlı olmadığı) çalışma kipini destekler.
- Launcher tabanlı etkileşim arayüzü ile kullanıcının günlük kullanım alışkanlıklarını değiştirmemektedir.
- Birçok mobil platformda, işletim sistemi tarafından sağlanan kullanıcı kişisel verilerine dair veri güvenliğinin uygulama seviyesinde sağlanması.
- Android mobil platformunda kullanıcı kişisel verilerini güvenli kullanımı için yenilikçi bir referans model oluşturmaktadır.

- İçsel olmayan bir uygulama biçiminde tasarlandığı için başka güvenlik çözümleri ile rahatça eşgüdümlü olarak çalışabilir, isteğe bağlı olarak dilendiğinde kaldırılabil

Kullanıcı kişisel bilgilerine yönelik ortaya çıkan güvenlik açıkları temel alınarak tasarlanan PIPSU mimarisinin, bu konuda günümüzde bulunan ve devamlı yenileri eklenen sorunlar gözönünde genişletilmesi ve zenginleştirilmesi konusunda birçok çalışma yapılabileceğigözlenmektedir. Bu yönde ileride yapılabilecek katkılar aşağıda belirtilmiştir:

- PIPSU uygulaması mobil marketlere sunulması : Bu tez çalışması kapsamındahedef mobil platform olarak seçilen Android'e yönelik bir uygulama olarak gerçekleştirilen PIPSU-L iOS, BlackBerry gibi diğer platformlar için de gerçekleştirilebilinir. PIPSU-L uygulaması ayrıca mobil marketlere konularak birincil kullanıcının geri dönüşleri alınarak geliştirilebilinir.

- PIPSU projesine özel SyncML tabanlı bulut servislerinin gerçeklenmesi: PIPSU platformunun halihazırdaki gerçeklenmesinde PIPSU-C bileşenleri için açık kaynak kodlu Funambol projesinin bulutundan yararlanılmıştır. İleriki versiyonlarda PIPSU'ya özel SyncML bulut servis kısımları geliştirilebilir.

- PIPSU-L içerisinde yerel olarak tutulan PIM kategorilerinin arttırılması: PIPSU mobil platform kullanıcısının telefon defteri ve takvim bilgilerinden oluşan PIM bilgilerinin güvenliğini sağlamaktadır. Bu kategori genişletilerek, mesajlar, notlar, vb kişisel bilgilerini de içermesi sağlanabilir.

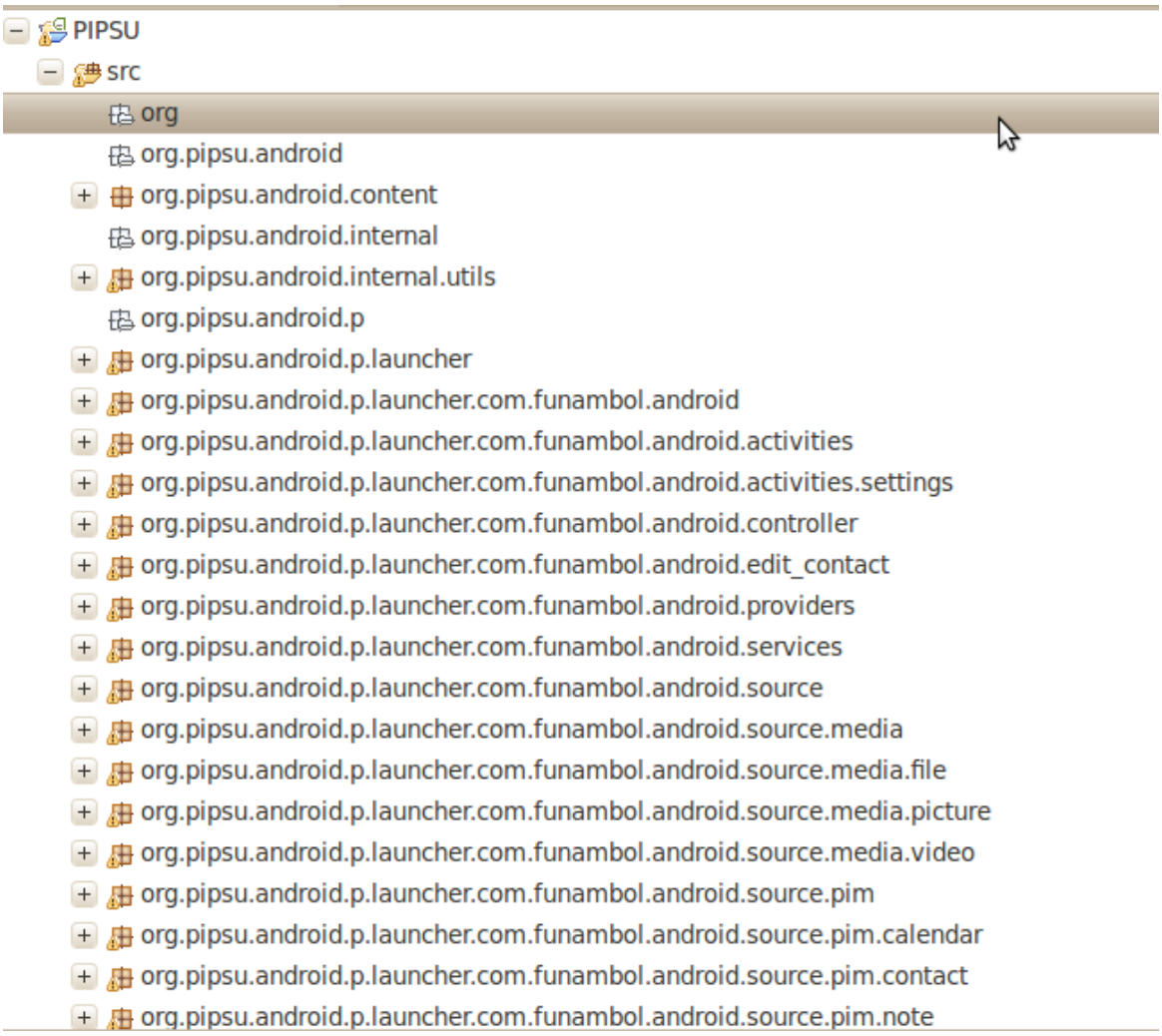
- PIPSU akranları arasında(P2P) DLNA veri paylaşımı: PIPSU platformu kullanılarak güvenli şekilde saklanan kullanıcı kişisel verileri; salt okunur kipte günümüz multi medya cihazları ile paylaşılabilir hale getirilebilinir. Örneğin günümüzün en yaygın dijital veri paylaşım protokolü olan DLNA[30] ile PIPSU-L'ye sahip cihazımızda bulunan kişisel verilerimizi televizyon veya multi medya oynatıcımız ile salt okunur kipte paylaşabiliriz.

- Üçüncü parti uygulamaların PIPSU altyapısını kullanabilmesi: PIPSU platformu ile üçüncü parti uygulamaların kullanıcı kişisel bilgilerini salt okunur modda kullanabilecekleri bir API sağlanabilir. Böylelikle kullanıcı verilerine erişimin güvenli ve teknokta üzerinden kontrolü sağlanmış olunur.

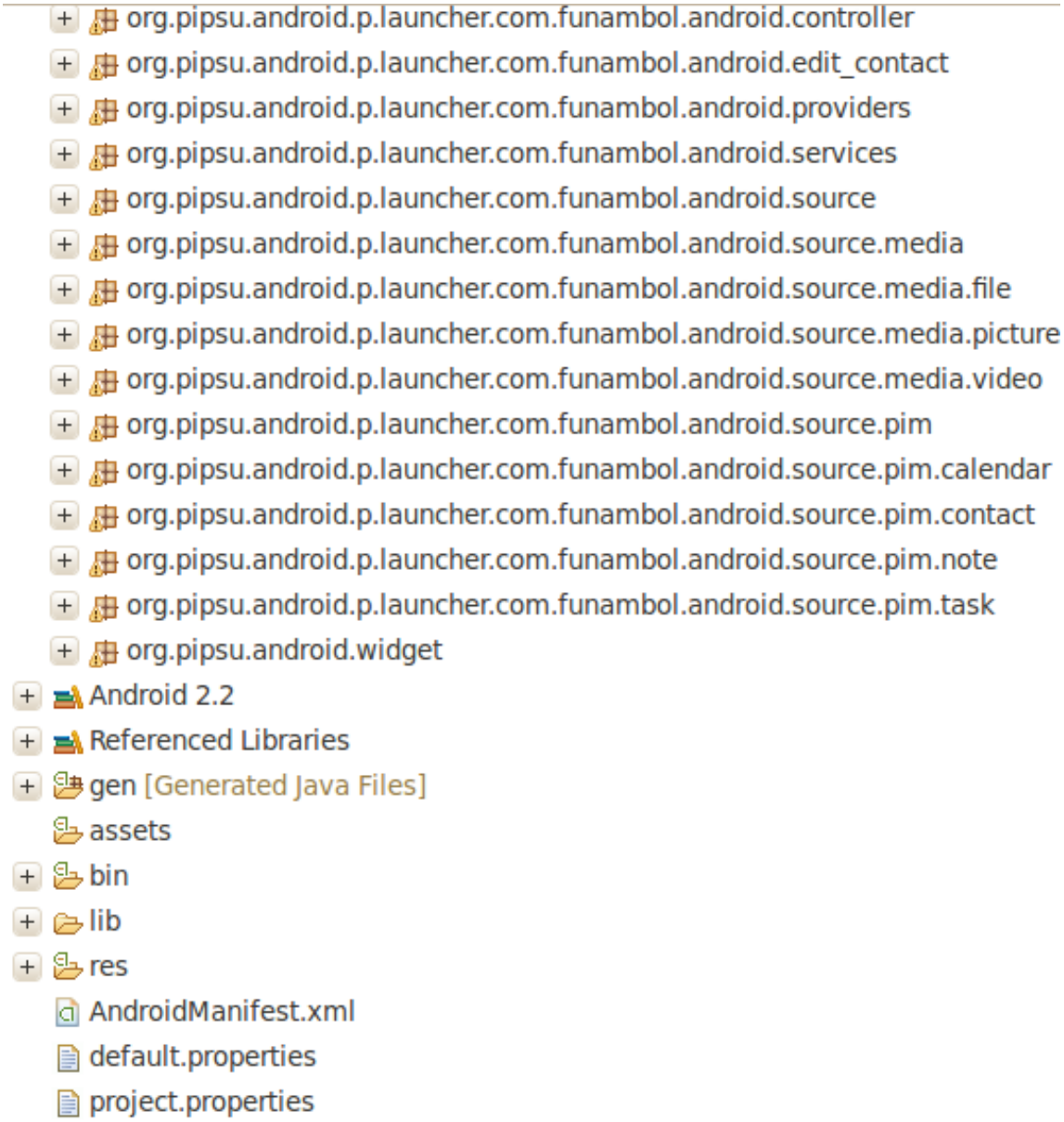
EK-A

KAYNAK KOD DİZİN YAPISI

PIPSU-L bir Android uygulaması olup, eclipse IDE(Integrated Development Environment)[32] aracılığı ile geliştirilmiştir. PIPSU-L geliştirimi Android SDK[31] aracılığı ile Android kütüphaneleri kullanılmış olup Android platform derleyicisi ve pakateleyicisi yine Android SDK dan alınarak yararlanılmıştır.



Şekil 0.1 PIPSU-L Kaynak Kodlarının Eclipse IDE'si üzerindeki görüntüsü



Şekil 0.2 PIPSU-L Kaynak Kodlarının Eclipse IDE'si üzerindeki görüntüsü(devam)

Şekil 0.2 ve Şekil 0.1'in devamı olup, Şekil 0.1 PIPSU-L uygulamasının Eclipse IDE'si package explorer görüntüsünü göstermektedir.

Aşağıda PIPSU Android uygulamasının dosya yapısı listelenmiştir.

Tablo 0.1 PIPSU-L Ana dizin yapısı

AndroidManifest.xml **assets** **default.properties** **lib** **project.properties** **res** **src** **./assets:** **./res:** **anim** **color** **drawable** **drawable-hdpi** **drawable-land-hdpi**	**drawable-land-mdpi** **drawable-ldpi** **drawable-mdpi** **drawable-port-hdpi** **drawable-port-mdpi** **layout** **layout-land** **layout-port** **values** **values-hdpi** **values-tr** **xml**

Tablo 0.2 PIPSU-L Launcher arayüz kaynak kod dosyaları

./src/org/pipsu/android/content: LauncherIntent.java LauncherMetadata.java ./src/org/pipsu/android/p: launcher ./src/org/pipsu/android/p/launcher: AddAdapter.java AllAppsGridView.java ApplicationInfo.java ApplicationsAdapter.java BubbleTextView.java CellLayout.java com DeleteZone.java DragController.java DragLayer.java DragScroller.java DragSource.java DropTarget.java FastBitmapDrawable.java FolderIcon.java FolderInfo.java Folder.java	HandleView.java InstallShortcutReceiver.java ItemInfo.java LauncherApplication.java LauncherAppWidgetHost.java LauncherAppWidgetHostView.java LauncherAppWidgetInfo.java Launcher.java LauncherModel.java LauncherProvider.java LauncherSettings.java LiveFolderAdapter.java LiveFolderIcon.java LiveFolderInfo.java LiveFolder.java ScreenIndicator.java ScreenLayout.java ScreenPrefActivity.java Search.java UninstallShortcutReceiver.java UserFolderInfo.java UserFolder.java Utilities.java Widget.java Workspace.java

Tablo 0.3 PIPSU-L Yardımcı kaynak kod dosyaları

./src/org/pipsu/android/p/launcher/com: funambol ./src/org/pipsu/android/p/launcher/com/funambol: android ./src/org/pipsu/android/p/launcher/com/funambol/android: activities AndroidAccountManager.java AndroidAppSyncSource.java AndroidAppSyncSourceManager.java AndroidConfiguration.java AndroidCustomization.java AndroidLocalization.java AndroidUtils.java AppInitializer.java	BuildInfo.java ConnectivityIntentReceiver.java ContactsImporter.java controller edit_contact ExternalAccountManager.java IntKeyValueSQLiteStore.java providers services SmsReceiver.java source StartupIntentReceiver.java SyncLock.java

Tablo 0.4 PIPSU-L Ayarlar arayüzü kaynak kod dosyaları

./src/org/pipsu/android/p/launcher/com/funambol/android/activities: AndroidAboutScreen.java AndroidActivitiesFactory.java AndroidAloneUISyncSource.java AndroidButtonUISyncSource.java AndroidDevSettingsScreen.java AndroidDisplayManager.java AndroidHomeScreen.java AndroidLoginScreen.java AndroidSettingsScreen.java AndroidSignupScreen.java AndroidUISyncSource.java settings	./src/org/pipsu/android/p/launcher/com/funambol/android/activities/settings: AndroidAdvancedSettingsTab.java AndroidDevSettingsUISyncSource.java AndroidSettingsTab.java AndroidSettingsUISyncSource.java AndroidSyncSettingsTab.java C2SPushSettingView.java SaveSettingsCallback.java SyncIntervallSettingView.java SyncModeSettingView.java TwoLinesCheckBox.java

Tablo 0.5 PIPSU-L Kullanıcı Hesap Modülü kaynak kod dosyaları

./src/org/pipsu/android/p/launcher/com/funambol/android/controller: AndroidAdvancedSettingsScreenController.java AndroidController.java AndroidHomeScreenController.java AndroidLoginScreenController.java AndroidSettingsScreenController.java AndroidSignupScreenController.java AndroidSyncModeHandler.java AutoSyncSwitcher.java	./src/org/pipsu/android/p/launcher/com/funambol/android/edit_contact: AndroidEditContact.java ContactDataStructure.java ContactValues.java EditContactDataView.java EditContactDateFieldView.java EditContactFieldView.java EditContactPhotoView.java EditContactView.java

Tablo 0.6 PIPSU-L PIM ve Multi medya soyutlama modülü kaynak kod dosyaları

./src/org/pipsu/android/p/launcher/com/funambol/android/source: AbstractDataManager.java AndroidChangesTracker.java media pim ./src/org/pipsu/android/p/launcher/com/funambol/android/source/media: file MediaAppSyncSourceConfig.java MediaAppSyncSource.java MediaExternalAppManager.java MediaIntentReceiver.java MediaSettingsUISyncSource.java MediaSyncSource.java MediaTracker.java picture video ./src/org/pipsu/android/p/launcher/com/funambol/android/source/media/file: AndroidFileObserver.java AndroidFileSyncSource.java FileAppSyncSourceConfig.java FileSettingsUISyncSource.java FileTracker.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/media/picture: PictureAppSyncSourceConfig.java PictureSettingsUISyncSource.java PictureSyncSource.java PictureTracker.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/media/video: VideoAppSyncSourceConfig.java VideoSettingsUISyncSource.java VideoSyncSource.java VideoTracker.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/pim: AndroidPIMCacheTracker.java calendar contact note PIMSyncSource.java PimTestRecorder.java task ./src/org/pipsu/android/p/launcher/com/funambol/android/source/pim/calendar: CalendarAppSyncSourceConfig.java	CalendarAppSyncSource.java CalendarChangesTracker.java CalendarChangesTrackerMD5.java CalendarExternalAppManager.java Calendar.java CalendarManager.java CalendarSettingsUISyncSource.java CalendarSyncSource.java EventSyncSource.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/pim/contact: AdditionalDataKinds.java ContactAppSyncSourceConfig.java ContactAppSyncSource.java ContactExternalAppManager.java Contact.java ContactManager.java ContactSettingsUISyncSource.java ContactsGroupsVersionCacheTracker.java ContactSyncSource.java DirtyChangesTracker.java FunambolContactManager.java Group.java GroupManager.java GroupVersionCacheTracker.java JoinedEnumerationContactsFirst.java VersionCacheTracker.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/pim/note: Note.java NoteSyncSource.java OINoteManager.java ./src/org/pipsu/android/p/launcher/com/funambol/android/source/pim/task: AstridTaskManager.java ./src/org/pipsu/android/widget: BoundRemoteViews.java ListViewImageManager.java SimpleRemoteViews.java WidgetCellLayout.java WidgetContentObserver.java WidgetCursorAdapter.java WidgetDataChangeListener.java WidgetListAdapter.java WidgetRemoteViewsListAdapter.java WidgetSpace.java

KAYNAKLAR

1. Nielsen; Ağustos 2011, Android pazar payı: http://www.dailytech.com/Android+Market+Share+Reaches+56+Percent+RIMs+Microsofts+Cut+in+Half/article22852.htm#
2. Apple Market ,Android Market, Black Berry Market Uygulama sayıları : Uygulama marketleri uygulama sayıları : Eylül 2011 http://148apps.biz/app-store-metrics/
3. Google Android:http://www.android.com/
4. M. Broersma. Serious security bugs found in Androidkernel.http://www.eweekeurope.co.uk/news/serious-security-bugs-found-in-android-kernel-11040, Nov. 2010.
5. The Android Project. What is android?http://developer.android.com/guide/basics/what-is-android.html.(2010-04-03).
6. Smartphones Too Smart For Your Own Good - Personal Information Leak Through Appshttp://m.ibtimes.com/smartphone-mobile-app-security-privacy-iphone-ios-android-personal-data-165619.html
7. Android Market a Breeding Ground for Malicious Mobile Appshttps://www.securityweek.com/android-market-breeding-ground-malicious-mobile-apps
8. Privacy consequences of using Android apps http://www.google.com.tr/url?sa=t&source=web&cd=2&ved=0CCMQFjAB&url=https%3A%2F%2Fwww.os3.nl%2F_media%2F2010-2011%2Fstudents%2Fberry_hoekstra%2Freport_priv_cons_using_android_apps.pdf%3Fid%3D2010-2011%253Astudents%253Aberry_hoekstra%253Assn%26cache%3Dcache&ei=fnH_TbCdC4TKswagoq3xDQ&usg=AFQjCNEFTsumHgAPVmVERQGN2BjImLhBfQ&sig2=iH-MdrVPSdCu-rURnsQNOQ
9. Global carriers take on Apple, Android with standardized app platformhttp://www.mobilemarketer.com/cms/opinion/editorials/5397.html
10. Google will regulate Android mobile processor standards :http://www.articlesbase.com/strategic-planning-articles/google-will-regulate-android-mobile-processor-standards-4588859.html#axzz1PqAVnO1K
11. A. Sabelfeld and A. Myers. Language-based information- flow security. IEEE Journal on selected areas in communi-cations, 21(1):5–19, 2003 ieeexplore.ieee.org/iel5/49/25986/01159651.pdf
12. SCanDroid: Automated Security Certification of Android Applicationswww.cs.umd.edu/~avik/papers/scandroidascaa.pdf
13. W. Enck, M. Ongtang, and P. McDaniel. Understanding Android security. IEEE Security & Privacy Magazine, 7(1):10–17, 2009.
14. W. Enck, M. Ongtang, and P. McDaniel. On lightweight mobile phone application certification. In CCS ’09: Computer and communications security, pages 235–245. ACM, 2009
15. M. Ongtang, S. McLaughlin,W. Enck, and P. McDaniel. Semantically Rich Application-Centric Security in Android. In ACSAC ’09: Annual Computer Security Applications Conference, 2009
16. Virtualized In-Cloud Security Services for Mobile Devices: jon.oberheide.org/files/mobivirt08-mobilecloud-pres.pdf
17. A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer,”Google Android: A comprehensive security assessment," Security Privacy, IEEE, vol. 8, no. 2, pp. 35-44, march-april 2010.
18. Monitoring Android for Collaborative Anomaly Detection: A First Architectural Draft : http://www.dai-labor.de/.../0808-02_DAI_TechReport_Monitoring_Android.pdf
19. Android Pazar payı: Nisan 2011, http://blog.nielsen.com/nielsenwire/?p=27418

20. Android mimarisi bileşen diyagramı http://elinux.org/images/c/c2/Android-system-architecture.jpg
21. Android uygulaması yaşam dongüsü http://t3.gstatic.com/images?q=tbn:ANd9GcQNoo-9m4X_oAIU0qvFMDxeBoWm3FAMuC3MjrxARcPE1g-6FAMI2A
22. Launcher+ :http://code.google.com/p/android-launcher-plus/
23. Funambol : http://www.funambol.org
24. APK : http://en.wikipedia.org/wiki/APK_%28file_format%29
25. ADB : http://developer.android.com/guide/developing/tools/adb.html
26. Gartner Haziran 2011 Market Raporu :http://www.gartner.com/it/page.jsp?id=1622614
27. IDC Mart 2011 araştırma raporu : http://www.idc.com/getdoc.jsp?containerId=prUS22762811
28. MobiThinking firmasının araştırma raporu : http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats
29. Linux Kullanıcı – dosya sistemi : http://www.kernel.org/doc/Documentation/filesystems/proc.txt
30. DLNA : http://www.dlna.org/
31. Android SDK : http://developer.android.com/sdk/index.html
32. Eclipse IDE : http://www.eclipse.org/

ÖZGEÇMİŞ

Adı Soyadı	Zana iLHAN
Adresi	Ovalbahçe Sitesi E.3.3 D:4 Merkez Mah./Sancaktepe İstanbul
Telefon	050503525141
E-posta	zanailhan@gmail.com

İŞ TECRÜBESİ

• Tarihler (başlangıç – bitiş)	06.2008 - Devam Ediyor
• Firmanın Adı ve Adresi	ARDIC A.Ş.
• İş tipi veya sektörü	Yazılım / ARGE
• Sahip olunan rol yada pozisyon	Yazılım Mimarı ve Cloud Takım Lideri
• Tarihler (başlangıç – bitiş)	04.2006-06.2008
• Firmanın Adı ve Adresi	NEVOTEK
• İş tipi veya sektörü	Yazılım / ARGE
• Sahip olunan rol yada pozisyon	Yazılım Uzm.
• Tarihler (başlangıç – bitiş)	01.2004 - 04.2006
• İşverenin Adı ve Adresi	IGE Ltd. Şti
• İş tipi veya sektörü	Yazılım
• Sahip olunan rol yada pozisyon	Yazılım Uzm.
• Tarihler (başlangıç – bitiş)	01.2003 - 01.2004
• Firmanın Adı ve Adresi	HVR TAMARA Ltd Şti
• İş tipi veya sektörü	Elektrik & Elektronik
• Sahip olunan rol yada pozisyon	Yazılım Uzm.
• Tarihler (başlangıç – bitiş)	01.2002- 01.2003
• Firmanın Adı ve Adresi	ITTS
• İş tipi veya sektörü	Elektrik & Elektronik
• Sahip olunan rol yada pozisyon	Yazılım Uzm.

EĞİTİM VE ÖĞRENİM DURUMU

Yüksek Lisans Okan Üniversitesi-Fen Bilimleri Enstitüsü, Bilgisayar 10.2009 - 01.2012Mühendisliği(Tezli)
LisansKocaeliÜniversitesi –Elektrik – Elekronik 5 YIL 09.2000 - 09.2007
LiseEskişehirYunusEmreAnadoluTeknik- Elektronik 06.1997

KURSLAR VE MESLEKİ GELİŞİM EĞİTİMLERİ

C veSistemProg. Derneği (C,C++, Java, PIC/ 8051, MFC, STL, ATL, SQL,)
CISCO (H323, SIP, Skinny)
Hand Held Products (J2ME, Java Micro Edition, Windows Mobile 2003 SE,

	CE 6.0) IBM Türkiye (UML2.0, RSA, Rational Rose, Purify)
SERTİFİKALAR	UML 2.0, C, C++ , Java, CISCO VoIP Telephony, ASF
DOĞUM TARİHİ/YERİ	10.11.1980-Doğubeyazıt / AĞRI
ASKERLİK DURUMU	Yaptı – ARALIK 2011
MEDENİ HALİ	Evli
SÜRÜCÜ BELGESİ	B
REFERANSLAR	**Necati ERGİN** C ve Sistem Programcıları derneği E-Mail:necatiergin@orsada.com **Kaan ASLAN** C ve Sistem Programcıları derneği E-Mail:aslank@csystem.org **Tunç KAHVECİ** ARDIC A.Ş. E-Mail:tunc.kahveci@ardictech.com **Haluk TÜFEKÇİ** ARDIC A.Ş. E-Mail: haluk.tufekci@ardictech.com **Mehmet AKSAYAN** ARDIC A.Ş. E-Mail: mehmet.aksayan@ardictech.com