



selcukcelik.org & https://www.youtube.com/user/socialmediagent



YAZILIM VE HİZMETLER

HAKKIMDA

HyperLedger Fabric (Kurumsal Blockchain)



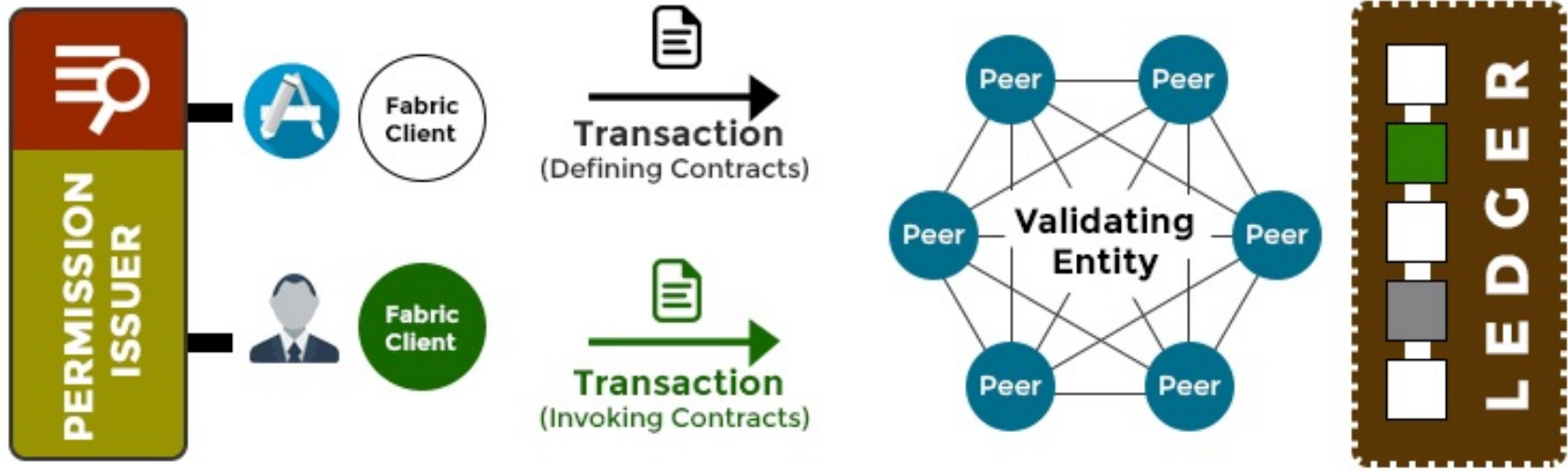
SELCUK CELİK

ENTERPRISE BLOCKCHAIN — 3 EKİM 2018





HYPERLEDGER FABRIC



HyperLedger Fabric (Kurumsal Blockchain) ve Çözümleri nasıl sağlarız?

Bu yazıyı okuduktan sonra ya ne kolaymış oturup hemen bizde bir çözüm yazalım demez iseniz bu yazının hiçbir anlamı kalmaz. Yani bu kadar kolay aslında kurumsal bir blockchain yapısına sahip olmanız. Ancak bunun olumlu ve olumsuz sonuçlarına da yazı içinde değineceğim. İlginç bilgiler vereceğim size. Siz bunları birleştirerek vizyoner bir paradigma ya geçeceksiniz.



Sizlerle daha önce kripto para tabanlı blockchain yazılarımı paylaşmıştım. Böyle bir yazılım sıfırdan yazan biri olarak da hiç saklamadan tüm yapıyı ve mimarisi ile çalışan halini video larım da anlatmıştım. Burada kurumsal blockchain yapılarına ve özellikle çok popüler olan HyperLedger şemsiyesi altında yer alan Hyperledger Fabric den bahsedeceğim.

Hyperledger Fabric Blockchain bize yapacağımız iş uygulamaları için low-level dediğimiz düşük seviye arabirimler sunar.

Yani örneğin akıllı kontrat kodları dağıtık sistemler üzerinde çalışır, yaptığımız girişler değiştirilemez deftere kaydedilir çıktılar ise veri saklama kısmında saklanır, işte yazacağımız uygulamalar bu düşük seviye kısmından soyutlama yani abstraction ile ayrılıp bu yapının en tepesinde yer alır. Soyutlama yani abstraction veri ve kontrolün soyutlanmasıdır. Daha altta yer alan sınıfların yani class ların içine bu sınıflardan türeteceğimiz hepsi de ortak özelliklere sahip sınıflar için boş veya dolu metod ve özelliklerin eklenmesidir. Biz bu abstract sınıftan kendi sınıfımızı türetir ve içinde yer alan metodları kullanır veya boş olarak konulan özellik veya metodları yeni türettiklerimiz içinde kendimiz kodlar doldururuz. Kısaca blockchain low-level altyapısında yer alan bu abstract yapı içindeki bizim için tanımlanmış ve bir çok işi yapan metodları kullanarak ve de türetip ekleyerek her işimizi gerçekleştirebiliriz.



Bu yapıyı gözümüzde çok basit canlandırmak için şöyle düşünün: Mimari yapıda en altta hyperledger fabric var (bize abstract yapı sunan low-level kısım) onun üstünde birazdan bahsedeceğim hyperledger composer ve en üstte iş uygulamaları var. 3 adet yapı yani.

Yukarıda en alt katmanı basitçe anlattım.

Şimdi **Hyperledger Composer** ile devam edeyim.Öncelikle composer ve hyperledger konusu içinde **business network** kavramını anlamanız lazım. Bu bildiğiniz bir business network değil. Onlar öyle isimlendirmişler. Çok basit anlamanız için bir yapı veya içinde bir yapıyı barındıran bir kutu olarak görün ve adı da business network. Biz bu yapının içinde **modelimizi**, **script (akıllı kontrat)** dosyalarımızı **ACL – Access Control List** dediğimiz erişim izinleri kontrol listesini ve **meta veriyi** saklarız.

Not: Hyperledger Fabric de consensus da bir değişiklik yapmak isterseniz başlangıç noktanız endorsement policy olacaktır. Bir ara o konuyu da detaylı anlatırım.

Diyelim hyperledger fabric de şirketimiz için bir çözüm yapmak istedik her çözüm için farklı bir business network (modeli) oluştururuz. Bu business network tanımları Business network archive da tutulur (dosya uzantısı bna dır). Yani iş çözümümüz ile ilgili modelimizi ve yani blockchain e yazılacak verilerimizin sınıflarının tanımlarını vb aynen OOP de olduğu gibi tanımlarız. Bunların içinde metodlar vb vardır. Bu



Modelin bir başka çok önemli noktası bu model tanımlandıktan sonra yani veri yapımız tanımlandıktan sonra sonraki aşamalarda **REST Server Api** özelliğini kullanarak bu tanımlanan model den otomatik olarak api oluşturulur ki bu harika işte dışarıdan biz bu api yi çağırarak verilerimizi blockchain e yazarız veya okuruz. Script içinde ise aslında akıllı kontrat kodu vardır ancak hyperledger dünyasında buna transaction processing function lar vardır.

Bu kodlar server da çalışır client tarafta değil bunu bilelim. Bir transaction olduğunda o transaction gidip bu ilgili script i bulur ve çalıştırır. Bu scriptleri composer da javascript ile yazarsınız. ACL ise örneğin siz çözümünüzde bu yazılımı kullanalardan şu kişi okuyabilir şu kişi okuyamaz göremez diyebilirsiniz. Dışarıdan giren kullanıcı da şunu yapar bunu yapamaz diyebilmeniz için aynı zamanda. Onları burada ayarlarsınız.

Bildiğiniz gibi hyperledger fabric kripto para destekleyen blockchainler gibi permissionless değil permissioned yapıdadır. ACL de bunun içindir. Metadata ise bu business network le ilgili isim, version, açıklama ve runtime esnasında yapı ile bu açıdan bağlantı kuracak bilgiler vardır.

Yukarıdaki yapıda bunları yapacak olan tek kişi de olabilir 2 ye de ayırabilirsiniz. Business network işlerini yazılım geliştirici ve hedef makinadaki yönetim ve deployment i da farklı bir admin yapabilir. Aksi halde ikisini de tek kişi yapabilir.



Bunları yaparken composer dan başka faydalı toolset ler var mı diye sorabilirsiniz?

Evet. **Web playground** ideal çözüm için business network tanımınızı çalıştırıp test eder vs vs, **Visual Studio Extension** var ve modelleme script vb orada yazabilirsiniz, **CLI** yani command line interface ki çok gerekli network ü deploy veya undeploy etmenizi, network archive oluşturmanızı, rest api oluşturmanızı vb sağlar..

Bir önemli nokta daha: Tek bir çözüm için tanımladığınız business network yapınız kendi docker i içinde çalışır ve bu da uygulamanıza izolasyon ve yönetim kolaylığı sağlar.

Lokal bilgisayarınızda composer i, cli i ve local repository yi (business network definition lar vb) oluşturabilirsiniz. (Yani yükleyebilirsiniz). Başlangıç noktanız bu olabilir. Sonra komut satırından docker ps komutu ile o an makinada hangi docker lar var bunu görebilirsiniz. 3 tane docker görmeniz mümkün. Biri hyperledger in member servisler container i ki hyperledger fabric e erişimi sağlayan vb olan docker, bir diğeri peer olan docker ki bu da bizim fabric e giriş noktamız, en başta yer alan docker da da bizim chain code yani akıllı kontrat vb ile bizim business network ümüzün yer alacağı docker biz burada çalışacağız. Burada **kitematic** diye bir tool var bunu kullanırsanız bu makinanızdaki docker vb her şeyi daha görsel ve kullanıcı dostu bir arabirimde görebilirsiniz. (aksi halde Linux komut satırından her şey).



Bu tool içinde bu 3 docker ı görmeniz mümkün. Bizim docker ımıza tıklarsanız container log ları göreceksiniz. Bu da içeride neler olduğunu görmeniz açısından güzel bir yapı.

Şimdi yeni çözümümüz ü tanımlamaya başlayabiliriz. Bir çok farklı yoldan buna başlayabilirsiniz. Tavsiyem playground u kullanarak başlamanız. Web tabanlı ve bu konularda geliştirme yapacağınız bir ortam sunuyor size. Zaten bu ekrana girdiğinizde web yani solda model file, script file ve ACL yi görebilirsiniz.

Bakın bu noktadan sonra işler ne kadar kolay oluyor inanamazsınız. Model dosyasına tıklayın ki cto uzantılı bir dosya bu. Sağda transaction larda yer alacak olan asset denilen veri yapınızın default değerleri var. İşte burada siz müşteri, hesap ve transaction gibi yapıları mevcut default değerleri değiştirerek kendi assetlerinizi oluşturmalısınız. Her asset içinde string tanımlar ve bazı yerlerde örneğin – – > Sampleasset asset diye bir yapıda görürsünüz. Çok basit asset Sampleasset tipinde demek bu. Sampleasset i de bu dosyada bir yerde class gibi tanımlamanız lazım. Bu yapıların tümü Model olarak geçer ve DSL dediğimiz domain specific language ile yazılır ki bunu aklınızda saklamanız gerekmez bilgi için verdim. Bu web editöründe syntax hatalarını dahi size içinde kod yazarken anında gösterir. Kod örneği resimdeki gibi olabilir size bir fikir vermesi için:



```
asset Account identified by accountId {  
  o String accountId  
  --> Customer owner  
  o Double balance  
}  
  
participant Customer identified by customerId {  
  o String customerId  
  o String firstName  
  o String lastName  
}  
  
transaction AccountTransfer identified by transactionId {  
  o String transactionId  
  --> Account from  
  --> Account to  
  o Double amount  
}
```

Bunu soldaki update butonu ile güncelleyin yani saklayın.

Sonra ekranın üstünde Define ve Test sekmelerinden Test i seçin. Gelen ekranda solda participants, assets ve transactions başlıklarında bu modelimizi göreceksiniz.

Bundan sonra yukarıdan Define tıklayıp biraz önceki model tanımlama kod ekranına gidip sol taraftan script file ı seçelim ve akıllı kodumuzu bu modele göre yazalım. (Bu bir javascript dosyasıdır.) Yapacağımız iş ilgili account lara para koymak. Sonra da birbirlerine bu paralardan bir kısmını göndermek. İşte aşağıdaki kodu:




```
/**
 * Sample transaction processor function.
 * @param {org.acme.sample.AccountTransfer} accountTransfer - the account
transfer
 * @transaction
 */
function accountTransfer(accountTransfer) {

    if(accountTransfer.from.balance < accountTransfer.amount) {
        throw new Error('Insufficient funds!');
    }

    accountTransfer.from.balance -= accountTransfer.amount;
    accountTransfer.to.balance += accountTransfer.amount;

    return getAssetRegistry('org.acme.sample.Account')
        .then(function (assetRegistry) {
            return assetRegistry.update(accountTransfer.from);
        })
        .then(function () {
            return getAssetRegistry('org.acme.sample.Account');
        })
        .then(function (assetRegistry) {
            return assetRegistry.update(accountTransfer.to);
        });
}
```

Buradaki kod atomic çalışır yani ya hep ya hiç. Bir de bu kod dağıtık yapıdaki peer (diğer blockchain consensus node bilgisayarları) ler de de çalışır ve o peer consensus sağlamazsa yani bir hata vb görürse yine veri doğru kabul edilmez transaction commit olmaz. Bunu da soldaki update ile güncelleyelim. Yani saklayalım.



Şimdi yine test e gidelim ve sağ üstte yeni bir katılımcı (müşteri) oluşturma butonu ile 2 müşteri oluşturup onlara para koyalım. Gelen ekran Json veri yapısındadır bilesiniz. Bizim model yapımızdaki customer ın saha verileri gelir ve oraya boş olan müşteri ID si, ad ve soyad ı girin. Alttaki Create New yani yeni oluştura basın ve işte blockchain de ilk müşteri veriniz oluşturuldu.. İkincisini de oluşturun. Bunları oluşturulduktan sonra ekranda da göreceksiniz.

Şimdi bu müşteriler için hesapları oluşturalım. Soldan Account u seçin. Ve aynı müşteri tanımladığınız gibi ve gelen ekranda dikkat edeceğiniz nokta müşteri ID si ne ise burada da owner ı aynı isimde olması. Balance a da para miktarını girin. Saklayın ve ikincisi için de aynısını yapın. Böylece iki müşteri ve 2 hesabımız var. Şimdi soldaki submit transaction butonuna tıklayın. Gelen ekran yine Json verisi olacak. From , to ve amount u girin ve submit e basın. Bu transaction registry ye gider. Bu da aynı zamanda blockchain de yer alır. Şimdi aynı ekranda soldaki account a basın gelen ekranda paranın bir hesaptan diğerine gittiğini göreceksiniz. Bu kadar kolay yani. Bu bizim iş modelimizin çalıştığını da gösterir aynı zamanda.

Buraya kadar işleri web tabanlı uygulamadan **interaktif** bir şekilde yaptık. Bu yaklaşımın bir başka faydası da danışman olarak gittiğiniz bir müşterinizde bunu bu şekilde kolayca gösterebilmeniz olacaktır. Bu oldukça fazla bir avantaj getirecektir size.



Şimdi bunu interaktif değilde lokalde oluşturduğum hyperledger fabric de çalıştırmak için neler yapmalıyız ona bakalım. Bunun için solda altta export seçeneği var. Bununla bu oluşturduğumuz model yapımızı ve tüm business network archive ı download edeceğiz. Oluşan dosya uzantısı bna olacaktır. Ve bunu lokal hyperledger fabric de çalıştıracamız tabii ilginç bir şekilde. Browser ımızın download klasöründe bu export dosyasını alıp lokaldeki hyperledger a import edeceğiz. Burada Composer ın Command Line utility sini kullanmanız gerekir. Composer network deploy komutu ve sağlayacağımız parametre değerleri ile bunu yaparız. Parametreler bu bna nın olduğu dizin ve enrollment hesap adı ile enrollment şifresi. Bu hesap adı ve şifresi makinalarınıza ilk anda lokal hyperledger kurarken oluşturulan hesap adı ve şifre olmalı. Ekranda deploy ediliyor biraz bekleyin mesajı gelir. Yaptığı iş şu: Bu business network tanımlarımızın bulunduğu yapının içinde yer alıp çalışacağı yepyeni bir docker oluşturmasından. Başta anlatmıştım her business network tanımları izolasyonu sağlamak için farklı docker larda çalışır. Aynı zamanda yönetiminde daha iyi olmasını sağlar bu yapı. Komut satırından docker ps –a yazarsanız bu oluşturulan docker ve tüm docker ları görürsünüz. Hatta kitematic uygulamasına giderseniz bu yeni oluşturulan docker ı görebilirsiniz.

Geldik en ilginç kısma. Bizim blockchain bilmeyen yazılımcılarımız var ve ister mobil ister web uygulamasında olsun onlara öyle bir kolaylık sunmalıyız ki bu uygulamaları kolayca hyperledger fabric e bağlasınlar. Yapacağımız iş Rest API yazmak ve bu yapıya bu API ler üzerinden bağlanıp onlara sağlanan bu API metodları ile



blockchain e bilgi yazıp okumalarını sağlamak. Bu rest api yi hatta biz hiç yazmayacak hyperledger fabric in bize sunduğu harika kolaylıklardan biri olan bu rest api yi bizim modelimize bakarak otomatik kendisinin oluşturmalarını sağlayacağız.

Komut satırından **composer-rest-server** yazın. Size bağlantı profilini sorar hyperledger fabric e bağlanmak için. Bu profiller default olarak home dizinindeki profile folder larında olur. Genelde ismi connection.json dosyası olarak geçer. Bunun içeriğine cat komutu ile bakarsanız burada bizim deploy ettiğimiz business network parametrelerini de görebilirsiniz. Başta bizden istediği bağlantı profili sorusuna defaultProfile diye girin. Business network identifier nedir sorusuna bizim buraya deploy ettiğimiz bna uzantılı dosyanın adını yazın. (connection.json daki isimde aynıdır.). Fabric username e bu makinaya hyperledger fabric i ilk kurarken set ettiğiniz username i girin sonra şifre ister onu da girin, gelen ekranda always use namespaces i seçin, Rest Api güvenli olsun mu sorusuna da N (no) diyebilirsiniz özellikle demolar için. Şunu belirtelim composer rest server plug edilebilir passport js authentication ını destekler ki bunu da bilmekte yarar var. Bunun faydası ne: Herhangi bir uygulama passport js i desteklerse onu burada rest api ye bağlanmak için kullanabilirsiniz demektir.

Bu sorulardan sonra sistem bizim business network tanımlamalarımızı discover yani keşif yaparak tespit eder. Şema , tip vb dosyalar oluşturup bize bir de rest api yi



çağıracağımız url adresini verir. Tabii esas Rest API yi oluşturur. Bize de bir iş bırakmaz. Bir de rest api yi yani metodları vb görmemiz için bize bir web adresi verir. Bu web adresine girin. Loopback API Explorer başlığı altında expose ettiği yani bize ve developer lara sunduğu api leri görebiliriz. Bunlardan birine tıklarsanız mesela account olana orada GET, POST, HEAD, PUT, DELETE diye REST komutlarını görebilirsiniz. Bunlardan her birinin yani örneğin GET veya POST a tıklayıp ilgili json ı düzenleyip test bile yapabilirsiniz. Yukarıda interaktif olarak yaptığımız hesaplara para koyma ve bir hesaptan diğerine para gönderme işini burada bu testleri yaparak da doğruluğuna bakabilir ve çalışıp çalışmadığını kontrol edebilirsiniz. Sonunda ise İşte bizim şirket içinde bu blockchain ile konuşacak desktop, web veya mobil yazılımlarını yazacak yazılım geliştiricilerimiz bu REST API leri çağırarak HyperLedger Fabric e bağlanıp oradaki modelimizle entegre kolayca olabilecektir. Bunu nasıl yapacaklarını yani bir Rest server üzerindeki API nin nasıl çağrılacağını ben burada anlatmayacağım ki onlar zaten bunu gayet iyi biliyorlar.

Bu aşamada hatta composer içinde Yeoman (kısaca Yo) temelli otomatik kod oluşturan şablonlar var. Komut satırından yo yazın ve ilgili istekleri seçin. (Hangi ortamda hangi tipte bir uygulama istiyorsunuz vb), şu an çalışan business network e mi bağlanmak istersiniz sorusuna Y deyin, uygulamanın adı ne olacak sorusunu yanıtlayın, uygulamanın açıklamasını girin, sahibi, email, business network identifier sorusuna bizim bna nın adını yazın, bağlantı profiline defaultProfile yazın, Enrollment Id ve Secret i default olarak bir şey girmeden enterleyin. Bu kadar. Eger



uygulamanın hangi tipte olacağı sorusuna AngularJS olsun dediyseniz buna göre gerekli olan tüm kodu modele bakarak üretiyor.

Hepsi bu kadar. !

Bugün bir çok kurumsal şirket , finans kuruluşları hatta bankalar içinde bu tür çalışmaları yapanlar var. Hatta yapıyı yazan ve paralel olarak mevcut sistemleri ile kullananlar var. Ama bir türlü tamam Ok geçelim bu sisteme diyemiyorlar. Güvenemiyorlar. Haklılar. Çünkü hyperledger fabric daha henüz olgun seviyede değil, yeterli ve derinine bilen insan kaynağı ve derinine destek verecek insan kaynağı yok. Veritabanında meydana gelecek bir patlağı ve verileri nasıl ne şekilde kurtaracakları, nerede ne olursa ne yaparız ın elde bilgisi yok. Benim de bir tavsiyem yok. Zamanla her şey yerine oturur. İnsanlar daha çok öğrensin ve onun için bunları durmadan yazıyor paylaşıyorum.

Buraya kadar yazdıklarımı harika uygulayan uzmanlar olabilir. Ancak blockchain uzmanı olmak başka bir şeydir. Bu kadarını yapanlar harika iş çıkarırlar. Ancak uygulama programcısı vb den öte olmaz. Blockchain uzmanı bana göre bir blockchain altyapısını yazan ve/veya gerçekten o altyapıyı çok iyi bilen ve yazabilecek potansiyelde olan insanlardır. Yani oturup bir hyperledger fabric gibi ya da ethereum gibi bir yazılımı yazan ya da yazabilecek altyapı ve donanımı olan insanlardan bahsediyorum. Bize hepsi lazım. Ve hepsini çok iyi bilen ve kullanan



insanların artması en büyük dileğim. Bir şeyler bilen biri olarak sizlere her zaman yapılan ve konuşma-temenni lerden ileri gitmeyen bize ve ülkemize hiçbir faydası olmayan insan ve kuruluşların oluşturduğu toplulukların anlattıkları yerine çok daha faydalı konuları ve teknolojileri bazan yüzeysel ama gerçek, bazan da derinine çok daha gerçek şekilde anlatmaya devam edeceğim. Konuşan yerine yapanları dinleyin. Tavsiyem bu.

Sağlıcakla kalın.

Selcuk Celik

SHARE



Bir Cevap Yazın

Comment as a guest.



Write your comment

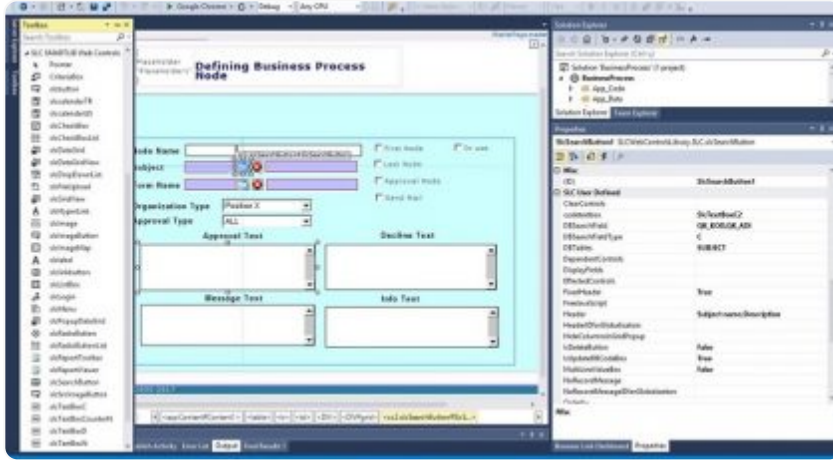
Name*

Email*

Post Comment



Read Next



Kurumsal Yazılım Framework ü Niçin ve Nasıl Yazılır.?

SELCUK CELİK ENTERPRISE FRAMEWORK — 3 EKİM 2018

Bazı üniversitelerimizde düzenlenen teknoloji etkinliklerinde geçmişte katılımcı olarak yer aldım. Fırsat bulduğumda ne öğrendiklerini ve nasıl uyguladıklarını aralarda öğrenmeye çalıştım.

...

Continue reading >



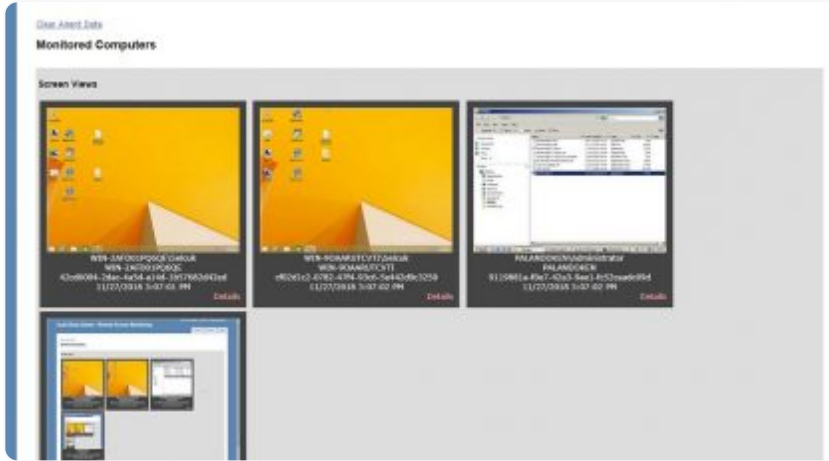
Araç Plaka Tanıma Yazılımları & CNN & HAAR & LBP

SELCUK CELİK PLATE RECOGNITION — 9 EKİM 2018

Plaka tanıma yazılımı konusunda bir çok denemelerim oldu. Özellikle resim üzerinden tanıma çok kolay.. İş video veya canlı video ya ...

Continue reading >





Remote Screen Monitoring – Uzaktan Ekran İzleme Yazılımı nasıl yazılır?

SELÇUK CELİK TEKNOLOJİ VE YAZILIM — 27 KASIM 2018

Bugün yine ilginç bir yazılım yaklaşımıyla karşınızdayım. Konumuz uzaktan bilgisayarları izleme vb yani Remote screen monitoring. İzlerken aynı zamanda mesaj ...

[Continue reading](#) >



Qualcomm-5G NR teknolojisi ve Apple

SELÇUK CELİK URUN INCELEME — 4 ARALIK 2018

Qualcomm un Hawaii deki etkinliğini özetleyeyim gerisi inanın önemli değil: Qualcomm haberleri aslında akıllı telefon kullanan herkesi ilgilendiriyor. Bu etkinlikte ...

[Continue reading](#) >

