



# ***TLOS***

---

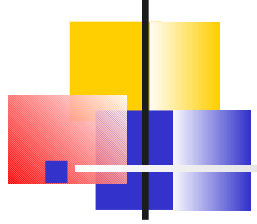
İş Yöneticisi  
v 1.0

Likya Bilgi Teknolojileri ve İletişim Hiz. Ltd.

# TLOS Nedir ?



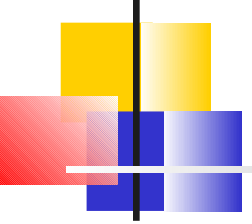
# TLOS Nedir ?



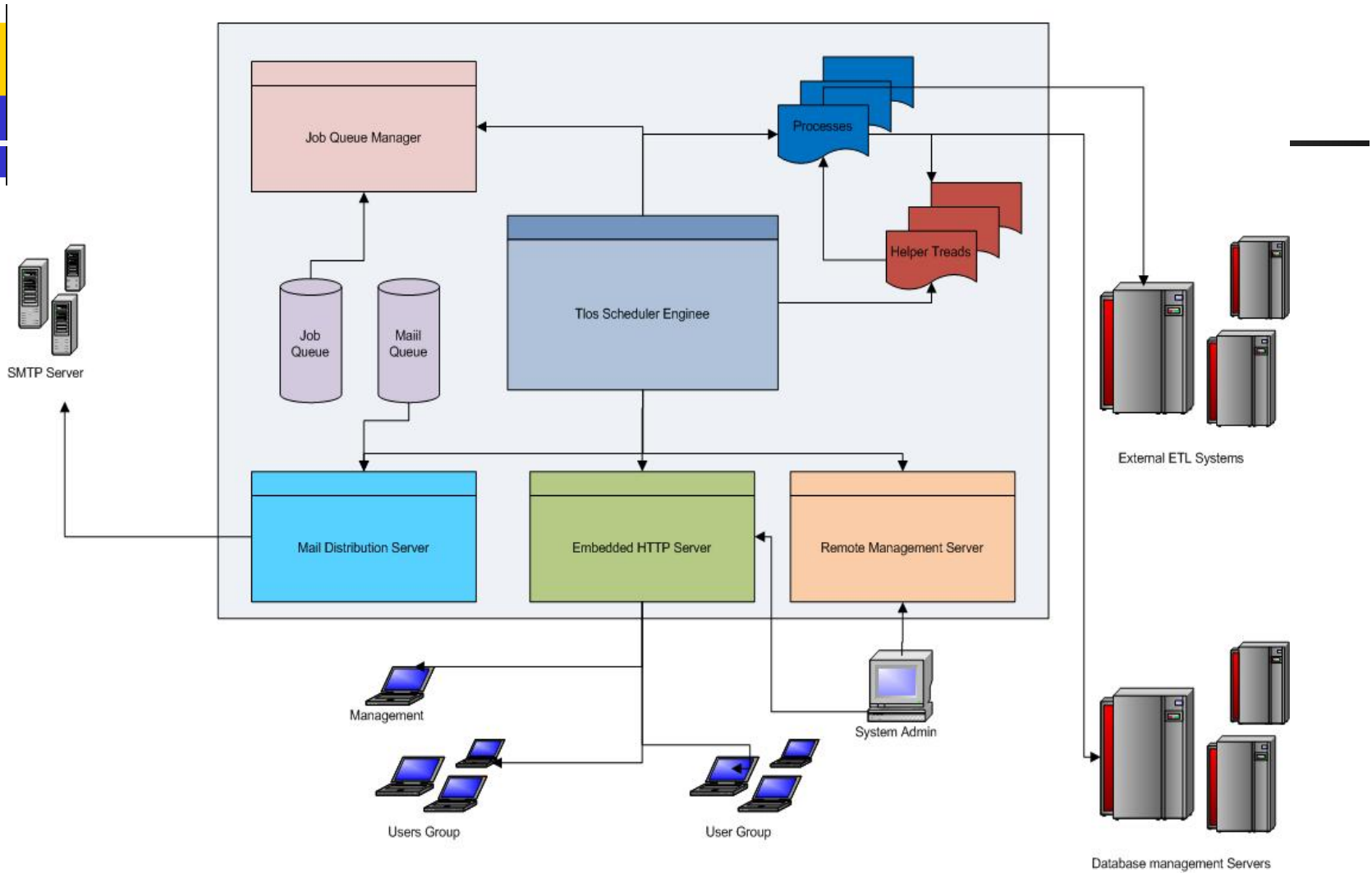
## ■ TLOS (Dalawa, Tlawa)

- Tlos Batı Lykia'nın Ksanthos Vadisi'nde bulunan kentlerindedir. *Kragoslar'ın (Akdağlar)* batı eteğinde, üç yanı sarp bir kaya-tepe üzerinde konumlanmıştır. Doğuda Patara kumsalına, batıda Kadyanda dağlarına bakmakta ve güneydoğusunda ise Pinara'yı görmektedir. Yapılan yüzey araştırmaları, egemenlik alanının başka hiçbir Lykia kentinde olmadığı kadar geniş bir coğrafyaya yayıldığını göstermiştir. Doğasıyla etkileyici olan bu benzersiz coğrafya, Akdağlar'dan çıkan sayısız kaynakla Anadolu'nun en bereketli topraklarına ve gemi yapımı için paha biçilmez sedir ormanlarına sahiptir. Tlos bu ayrıcalıklı konumu sayesinde Lykia'nın en erken, en önemli ve en zengin kentlerinden biri olmuş; bu özel yerini, Lykia Birliği'nin üç oy hakkına sahip altı kentinden biri olarak da tarih boyunca korumuştur.

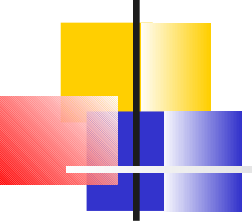
# ***TLOS*** Nedir ?

- 
- 
- Tanımlanan işleri verilen zaman ve bağımlılık kuralları çerçevesinde çalıştıran, yöneten ve loglayan platform bağımsız bir iş yönetim sistemi

# Tlos Scheduler Architecture



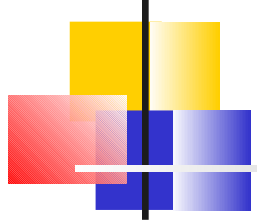
# ***TLOS*** Neler Sunuyor ?

- 
- 
- Platformdan bağımsız çalışma ortamı
  - Akıllı Yük Yönetimi (Loadbalancing)
  - Kolay kullanılabilir tanım dosyası
  - Detaylı senaryo tanımları
  - Komut satırından yönetim
  - Platform ve lokasyon bağımsız yönetim ve operasyon web arabirimi

# ***TLOS*** Neler Sunuyor ?

- Paralel senaryo alıřtırma
- Kaldığı yerden süreci devam ettirebilme yeteneđi
- Detaylı ve özet loglama alt yapısı
- Hata ve uyarı ayrımı yapılması
- e-posta özelliđi

# Platformdan bağımsız çalışma ortamı



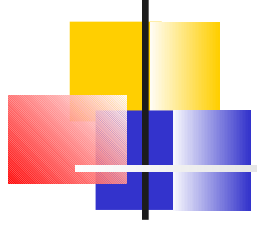
Java dili ile geliştirilen Tlos uygulaması, belirtilen Jdk sürümünün (1.6) işletim sistemi tarafından desteklenmesi durumunda, her türlü platforma çalışacak şekilde tasarlandı.

Şu ana kadar, Windows Vista, Windows XP, HP-UX ve Linux Fedora platformları üzerinde başarı ile test edildi.

# Akıllı Yük Yönetimi (Loadbalancing)

- Anlık çalışan iş sayısının alt ve üst limitlerinin tanımlanabilmesi
- Scheduler işler üst limite ulaştıktan sonra alt limite düşene kadar yeni iş yüklemes
- Böylelikle kullanılan sistemlerin etkin kullanımı ve aşırı yüklenmemesi sağlanmış olur

# Kolay kullanılabilir tanım dosyası



Örnek tanım dosyası :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>
```

ispersistent: Programın herhangi bir şekilde sonlanması durumunda, son halini diskten yükleyip tekrar başlayabilir.

usemail: E-posta sistemi kullan/kullanma, bu parametre false olur ise, diğer e-posta parametreleri okunmaz ve Tlos tarafından e-posta üretilmez.

statusListForMail: Biten işlerin bitiş sonuçlarına göre, rapor e-postası gönderilmesi istenen durumlar virgülle ayrılarak bu kısma yazılır.

Örnek iş sonuçları : READY,WORKING,WAITING,SUCCESS FAIL,TIMEOUT

emaildestination: Sistem raporlarının gönderileceği e-mail adresi

emailusername: SMTP server için gerekli olan kullanıcı adı

emailpassword: SMTP server için gerekli olan şifre

smptserver: SMTP server adresi

scenario: Senaryo bilgilerinin alınacağı dosya

logfile: log dosyası

schedulerFrequency: Scheduler'ın çalışma sıklığı (sn)

schedulerHigherThreshold: Tlos'un kullandığı akıllı yük dengeleme algoritması için gerekli olan havuz üst limiti.

schedulerLowerThreshold: Havuz alt limiti.

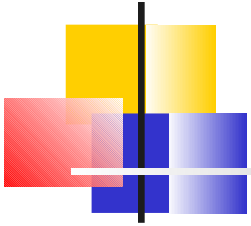
managementPort: Komut satırı yönetim konsolu haberleşme portu

managementBufferSize: Komut satırı yönetim konsolu buffer boyu

httpHostName: Web arabirim için, Tlos'un bulunduğu sistemin ip adresi

httpAccessPort: Web arabirimi çalışması için kullanılacak TCP portu.</comment>

# Kolay kullanılabilir tanım dosyası



```
<entry key="ispersistent">>true</entry>
<entry key="usemail">>false</entry>
<entry
  key="statusListForMail">READY,WORKING;WAITING,SUCCESS,WAITING,TIMEOU
  T</entry>
<entry key="emaildestination">likya@likyateknoloji.com</entry>
<entry key="emailusername">likya@likyateknoloji.com</entry>
<entry key="emailpassword">password</entry>
<entry key="smtserver">mail.likyateknoloji.com</entry>
<entry key="scenario">senaryoBilgileri.xml</entry>
<entry key="logfile">TlosEkran.log</entry>
<entry key="schedulerFrequency">2</entry>
<entry key="schedulerHigherThreshold">8</entry>
<entry key="schedulerLowerThreshold">4</entry>
<entry key="managementPort">3001</entry>
<entry key="managementBufferSize">128</entry>
<entry key="httpHostName">localhost</entry>
<entry key="httpAccessPort">3000</entry>
</properties>
```

# Detaylı senaryo tanımlanabilmesi

Senaryo bilgileri, xml tabanlı bir tanım dosyası içinde bulunmaktadır.

Örnek senaryo dosyası :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>
    Senaryo bilgilerini bulunduran dosya
    jobname=
    command,
    type,
    log_file_path,
    dependencylist,
    result,
    time (gg:aa:yyyy hh:mm:ss) veya (hh:mm:ss başladığı günü kabul edip çalışır)
    ,timeout(sn)
    ,autorestart (evet, hayır)
  </comment>
  <entry key="job1">notepad.exe,system,/usr/lib,yok,S,22/05/2008 15:02:00,10,evet</entry>
  <entry key="job2">calc.exe,system,yok,job1|job5,S,22/05/2008 15:02:00,20,hayır</entry>
  <entry key="job3">calc.exe,system,yok,job2,S,22/05/2008 15:02:00,20,hayır</entry>
  <entry key="job4">calc.exe.bat,system,yok,job2,S,22/05/2008 15:02:00,20,hayır</entry>
  <entry key="job5">calc.exe,system,yok,yok,S,22/05/2008 15:02:00,20,hayır</entry>
  <entry key="job6">calc.exe,system,yok,job3|job4,S,22/05/2008 15:02:00,20,hayır</entry>
</properties>
```

# Detaylı senaryo tanımlanabilmesi

Örneğin :

```
<entry key="job21">  
    job21.sh,system,/usr/lib,job12,S,22/05/2008 15:02:00,10,evet  
</entry>
```

Yukarıdaki tanıma göre job21 tarih 22/05/2008 ve saat 15:02:00 olduğunda çalıştırılacak.

Verilen time-out süresi aşırsa ve autorestart evet ise Job timeout'a düşüp fail olacak ve bir kerelik restart edilecek, autorestart hayır ise statu timeout olup job bir timeout süresi kadar daha beklenecek bitmezse fail edilecek.

# Komut satırından yönetim

**Tlos uygulaması, TlosRemoteManager komut satırı uygulaması yardımı ile çalışma esnasında yönlendirilebilmektedir.**

## Koşullu kapanış

- Çalışmakta olan joblar biter bitmez, bir sonraki adıma geçmeden uygulama kapatılır.

- `java -cp LikyaTlos.jar com.likya.tlos.TlosRemoteManager -port 3001 -host localhost -terminate`

## •Koşulsuz kapanış

- Çalışmakta olan uygulama, bütün jobları ile birlikte hemen kapatılır, devam eden işler ise yarıda kesilir.

- `java -cp LikyaTlos.jar com.likya.tlos.TlosRemoteManager -port 3001 -host localhost -fterminate`

## •Tekrar deneme

- Fail statüsünde bulunan bir job istenirse, tekrar çalışması sağlanır.

- `java -cp LikyaTlos.jar com.likya.tlos.TlosRemoteManager -port 3001 -host localhost -jobname xxxx`

# Platform ve lokasyon bağımsız yönetim ve operasyon web arabirimi

- Tlos uygulaması, web üzerinden sağladığı yönetim ve bakım konsolu sayesinde, uzaktan erişime imkan sağlamaktadır.
- Çalışan işlerin anlık durumlarını görüntüleyebilir, ve uzaktan erişim modülünün sağladığı tüm imkanlarıda kullanıcıya verir.
- Güvenli erişim arabirimi, uygulamanın ve sürecin kontrol ve yönetimini oldukça kolaylaştırmaktadır.

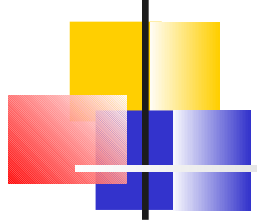
# Platform ve lokasyon bağımsız yönetim ve operasyon web arabirimi

---

Web arabirim uygulaması, Tlos içine gömülü olduğundan ayrıca ek bir operasyonel yük ve bağımlılık getirmemektedir.

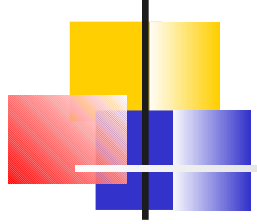
- Web arabirimi, sadece kullanıcı adı ve şifre ile kullanılabilir, bu sayede güvenli bir kontrol sağlanmaktadır.

# Paralel iŖ ve sre alıŖtırma



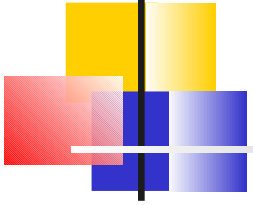
Tlos uygulaması, birbirinden bağımsız iŖ ve sreleri aynı anda paralel olarak alıŖtırıp ynetme imkanına sahiptir, burada sınırlar, donanım zellikleri tarafından belirlenmektedir.

# Kaldığı yerden süreci devam ettirebilme yeteneđi



Tlos uygulaması, eđer tanım dosyasında tanımı yapıldı ise, koşullu ya da koşulsuz kapatılma ya da dışarıdan “kill” ile sonlandırma durumlarında, Tlos.recover dosyası içinde tuttuđu yaşam bilgileri sayesinde, yeniden çalıştırıldığında, kapanmadan önceki son durumu başlangıç kabul edip, sürece veya süreçlere kaldığı yerden devam edebilme yeteneđine sahiptir.

# Detaylı ve özet loglama altyapısı



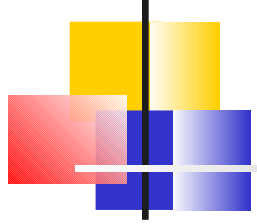
Tlos uygulaması, detaylı ve özet olmak üzere, iki ayrı log dosyası üretmektedir.

Bunlardan özet olanı, ekrana yazılan logları, detay olanı ise, her türlü işlem detayını yansıtmaktadır.

Bu sayede, istenildiğinde log dosyaları incelenerek süreçlerle ilgili bir çok detay bilgiye ulaşılabilir.

Ayrıca, hata durumları da detaylı olarak gözlemlenebilir.

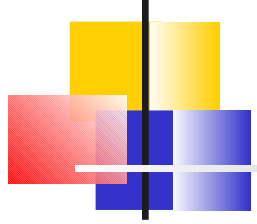
# Hata ve uyarı ayrımı



Tlos uygulaması, başarı ile biten uygulamaları, uyarı vererek bitenler ile tamamen hatasız bitenler olmak üzere ayırıp, görüntüleyebilmektedir.

Bu sayede, istenildiğinde log dosyaları incelenerek, süreci etkilemeyen ancak sonucu etkileyebilecek hatalar önceden farkedilip giderilebilir.

# E-posta uyarıları



Tlos uygulaması, tanımları yapılmak koşulu ile, işlerin bitiş raporlarını ve sonuçlarını, tanımlı e-posta adresine gönderebilmektedir. Bu sayede, sonuçlar anlık olarak takip edilebilirken, ayrıca uyarı e-postaları sayesinde de takip edilip ve gözlemlenebilmektedir.



Teşekkürler...

Likya Bilgi Teknolojileri ve İletişim Hiz. Ltd.

[tlos@likyateknoloji.com](mailto:tlos@likyateknoloji.com)

