Overview

Unity Database Security

In-Rest DB Encryption TDE + In-Transit TLS encryption + Enhanced Database Backup

THE AIRTIME

Mobileware Group

Database Security

Protecting satellite customer data is essential - making sure that data stay intact and that the typically high profile customer data are protected from any type of external unwanted access.

Their are many parts to be considered when it come to protecting data, across system architecture, data transport protection, core data protection in the databases as well as the protection of the actual database backup’s.

Design a Secure System

Encrypt Essential Data

Enhanced Firewalls

Database Security

Protecting the core data in Unity can be applied using different security levels:

Database Encryption: Database encryption on MariaDB will be possible with TDE
Data Transmit Protection: Data between clients + between databases with TLS
Dedicated Firewalls: Each Database node protected with Firewal filters
End to End Data Protection: SSL/TLS end to end from user to database
Database Backup: Database backup that support encrypted data
Database Access Policies: No Direct External Internet access
Database Table Audit: Log + Record user access + data changed (old/new values)
Database Security + OS Patches: Apply regulary recommended patches

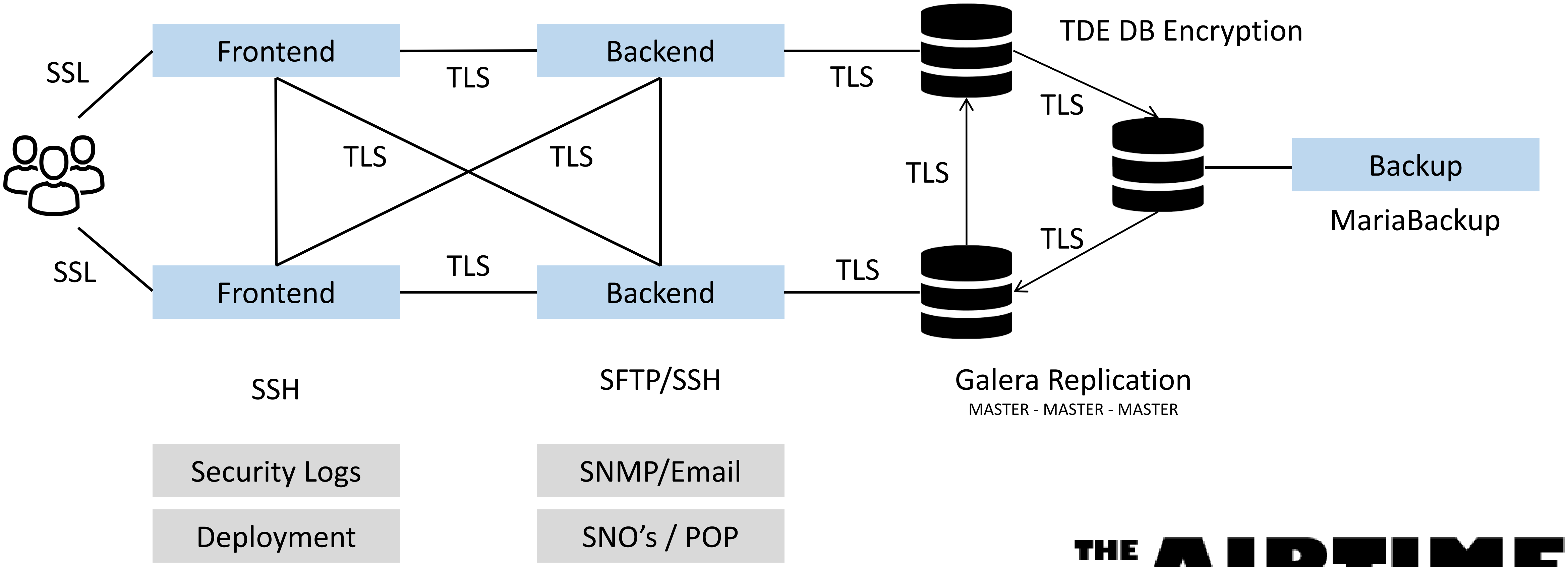
Unity Architecture

Redundant - Highly Scalable Architecture that apply highest possible level of Security

THE AIRTIME

Mobileware Group

Unity Architecture



SSL

SSL

Frontend

Frontend

TLS

TLS

TLS

TLS

Backend

Backend

TLS

TLS

TDE DB Encryption

TLS

TLS

TLS

Backup

MariaBackup

Galera Replication

MASTER - MASTER - MASTER

SSH

SFTP/SSH

Security Logs

Deployment

SNMP/Email

SNO’s / POP

Transparent Data Encryption (TDE)

TDE Encrypt Data stored in the database + Encrypt sepected log files to prevent unwanted access

Transparent Data Encryption (TDE)

In a normal database setup will unwanted access grant access to stored data through standard database setup as well as logfiles and server history.

With the later versions of MariaDB is it possible to encrypt “In-Rest” data.

One solution is to encrypt sensitive data in a database and use a certificate to protect the keys that encrypt the data. This solution prevents anyone without the keys from using the data. But you must plan this kind of protection in advance.

TDE does real-time I/O encryption and decryption of data and log files. The encryption uses a database encryption key (DEK). The database boot record stores the key for availability during recovery. The DEK is a symmetric key. It's secured by a certificate that the server's master database stores or by an asymmetric key that an EKM module protects.

TDE protects data at rest, which is the data and log files. It lets you follow many laws, regulations, and guidelines established in various industries. This ability lets software developers encrypt data by using AES and 3DES encryption algorithms without changing existing applications. Encryption of a database file is done at the page level. The pages in an encrypted database are encrypted before they're written to disk and are decrypted when read into memory. TDE doesn't increase the size of the encrypted database. Their is still full access to data using normal SQL commands - from accounts that have the needed permission to access data.

Transparent Data Encryption (TDE)

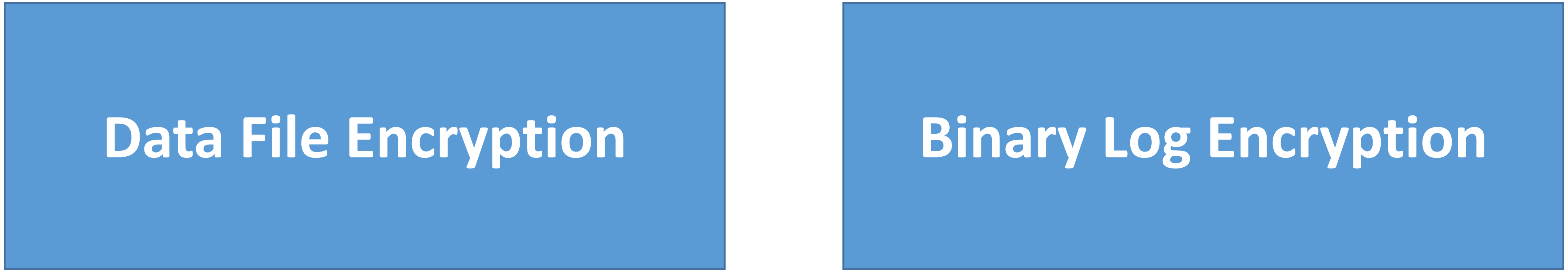
Unity allow its Galara Replicated Database structure to be TDE encrypted to prevent unwanted user access that have been able to get through the normal Firewall protection.

Data are incrypted using in internal generated token - in the case data need to get restored then will this be possible using the correctset of keys.

Data encryption take place on each single server - meaning data isn’t protected when in transit beween external clients as well as between the different nodes in our Galara Database Replication setup - In-Transit data protection are handled by installing TSL between client server / database servers as explained in the next part.

Transparent Data Encryption (TDE)

Database Encryption consist of two main area’s.



Unity encrypt the key files but leave certain files used for maintenance unencrypted like example the slow query file and similar.

In Transit Data Protection (TLS)

Protecting Data being send between server (Database servers / Clients) using TLS tunneling

THE AIRTIME
Mobileware Group

In-Transit Data protection (TLS)

Communication between Client Servers (Backend Servers) and Unity Database environment can be protected using TLS tunneling - that will ensure that transit data is protected end to end from ‘man in the middel’. When Unity is installed in closed network environments will this be optionally as long as the operators own security can enforce the needed data protection.

MariaDB supports encrypted connections between clients and the server using the TLS (Transport Layer Security) protocol. TLS is sometimes referred to as SSL (Secure Sockets Layer) but MariaDB does not actually use the SSL protocol for encrypted connections because its encryption is weak.

In-Transit Data protection (TLS)

Database In-Transit consist of two main area’s in Unity.

Client-Server Encryption

Replication Encryption

Encryption certificate is using our CERT wildcard *.theairtime.com and in certain cases an internal self signed OpenSSL certificate all in PEM format. Client-Server part is our Backend nodes and the replication is between our Galara Database Master - Master - Master communication.

Web Data Protection (SSL)

Entering or listing Data stored in the Database Environment through Frontend Environment

THE AIRTIME

Mobileware Group

Web Data protection (SSL)

Data stored in the database environment can be accessed through Unity - meaning this channel need to be as well protected as the backend data stored in the database enviroment. Unity have stand alone web server in a highly hardend setup that parse stateless request back to the backend environments - meaning absolutly no data is stored in frontend servers - these are ‘only’ applying layout, colors and similar to the data being processed.

Data between end user and frontend server are protected with a standard SSL certificate with a key size of 2048 in RSA mode. Data send from and to frontend server using our Unity API parser will as well between backend and frontend be running through a TLS connection to the backend servers.

Only port 443 is open in direction of Internet and this is applyed with automated banning software.

Unity Database Backup

Server Image Backup Model + Incrementral Backup Model

THE AIRTIME

Mobileware Group

Unity Database Backup

Unity is operated with two types of backup - Server backup that periodically take a complete snap shoot of the backend + database servers - this image snap shoot model allow extrem fast server recovery compared to loading traditional database backups that can take extrem many hours. Server snap shoots will all have database data encrypted.

The other model os standard incremental database backup - here are being applied a major change linked to database encryption - our standard Percona Xtrabackup that currently are being used do unfortunatly not support database encryption and table compression - so to support this will a new backup be needed annd this will be based on MariaBackup that offer the needed functionality.

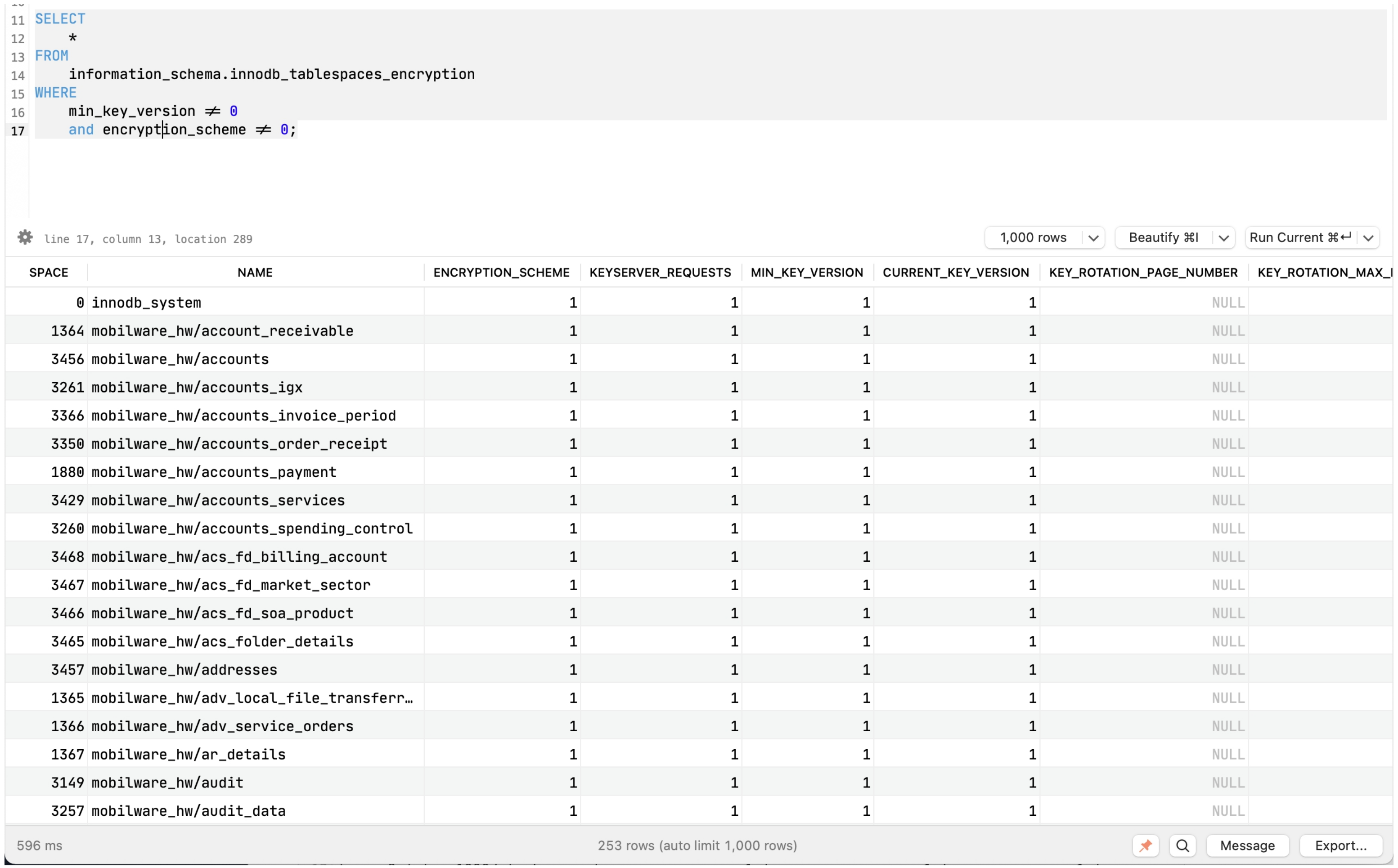
Unity DB Encrypt / TLS

After Encrypt + TLS

THE AIRTIME

Mobileware Group

Unity Database Encryption - In-Rest



```
SELECT
    *
FROM
    information_schema.innodb_tablespaces_encryption
WHERE
    min_key_version ≠ 0
    and encryption_scheme ≠ 0;
```

line 17, column 13, location 289

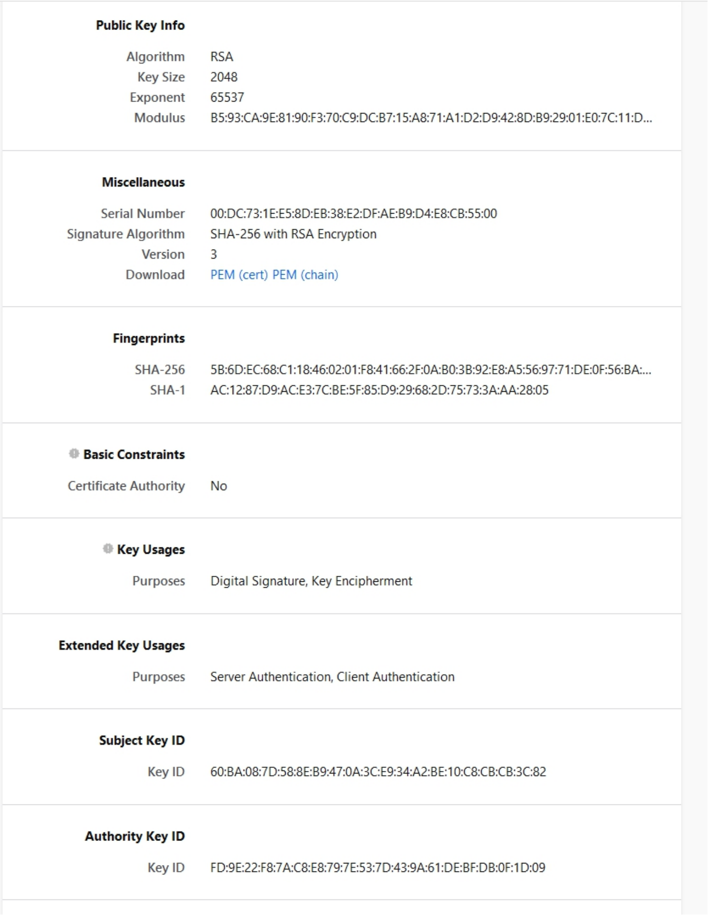
SPACE	NAME	ENCRYPTION_SCHEME	KEYSERVER_REQUESTS	MIN_KEY_VERSION	CURRENT_KEY_VERSION	KEY_ROTATION_PAGE_NUMBER	KEY_ROTATION_MAX_I
0	innodb_system	1	1	1	1	NULL	
1364	mobilware_hw/account_receivable	1	1	1	1	NULL	
3456	mobilware_hw/accounts	1	1	1	1	NULL	
3261	mobilware_hw/accounts_igx	1	1	1	1	NULL	
3366	mobilware_hw/accounts_invoice_period	1	1	1	1	NULL	
3350	mobilware_hw/accounts_order_receipt	1	1	1	1	NULL	
1880	mobilware_hw/accounts_payment	1	1	1	1	NULL	
3429	mobilware_hw/accounts_services	1	1	1	1	NULL	
3260	mobilware_hw/accounts_spending_control	1	1	1	1	NULL	
3468	mobilware_hw/acs_fd_billing_account	1	1	1	1	NULL	
3467	mobilware_hw/acs_fd_market_sector	1	1	1	1	NULL	
3466	mobilware_hw/acs_fd_soa_product	1	1	1	1	NULL	
3465	mobilware_hw/acs_folder_details	1	1	1	1	NULL	
3457	mobilware_hw/addresses	1	1	1	1	NULL	
1365	mobilware_hw/adv_local_file_transferr…	1	1	1	1	NULL	
1366	mobilware_hw/adv_service_orders	1	1	1	1	NULL	
1367	mobilware_hw/ar_details	1	1	1	1	NULL	
3149	mobilware_hw/audit	1	1	1	1	NULL	
3257	mobilware_hw/audit_data	1	1	1	1	NULL	

596 ms

253 rows (auto limit 1,000 rows)

THE AIRTIME
Mobileware Group

Unity Database TLS - In-Transit



Public Key Info	
Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	B5:93:CA:9E:81:90:F3:70:C9:DC:B7:15:A8:71:A1:D2:D9:42:8D:B9:29:01:E0:7C:11:D...
Miscellaneous	
Serial Number	00:DC:73:1E:E5:8D:EB:38:E2:DF:AE:B9:D4:E8:CB:55:00
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	PEM (cert) PEM (chain)
Fingerprints	
SHA-256	5B:6D:EC:68:C1:18:46:02:01:F8:41:66:2F:0A:B0:3B:92:E8:A5:56:97:71:DE:0F:56:BA:...
SHA-1	AC:12:87:D9:AC:E3:7C:BE:5F:85:D9:29:68:2D:75:73:3A:AA:28:05
Basic Constraints	
Certificate Authority	No
Key Usages	
Purposes	Digital Signature, Key Encipherment
Extended Key Usages	
Purposes	Server Authentication, Client Authentication
Subject Key ID	
Key ID	60:BA:08:7D:58:8E:B9:47:0A:3C:E9:34:A2:BE:10:C8:CB:CB:3C:82
Authority Key ID	
Key ID	FD:9E:22:F8:7A:C8:E8:79:7E:53:7D:43:9A:61:DE:BF:DB:0F:1D:09

THE AIRTIME
Mobileware Group

Unity Elements

Basic Security setup linked to the different elements in a normal system setup

THE AIRTIME

Mobileware Group

Unity HA Environment

Frontend HA: Two Frontend servers will be installed that both are active and DNS balancing can be used to distribute load – round robin with monitor – Airbus DS DNS.

Frontend can automatically swap between backend nodes based on availability

Operator Dashboard HA: Same as Frontend – two seperate servers operating that automatically can swap from backennd to backend.

Database HA: Database environment based on MariaDB with a version that support TDE Data Encryption (tables/logfiles etc.) Galera Replicated Database environment with Three Servers running Master-Master-Master mode.

Backend HA: Two backend nodes are installed ontop of the HA Database environment – both are operational – but frontends and dashboards will only be using primary until a failover take place.

Database Backup: Database and server backup based on server image snapshot that are taken periodically to allow ‘very’ fast recovery + normal database backup based on MariaBackup that support Encrypted Database format + Compressed Table handling - replacing our earlier XtraBackup from Percona.

Unity Frontend Basic Security

Frontend Environment: This is basically a plain web server with absolutly no user data stored – it is only delivering layout based on standard Bootstrap CSS – and multiple Frontends in different style can be implemented.

Frontend connectivity: Only SSL is available from outside – and all connectivity to backend environment are running through SSL tunnel parsing our web service API communication. All is stateless with authentication on each single request

Frontend security: Frontend environment are getting scanned periodically using HP Fortify, OWASP scanner, have integrated login scanner with auto banning, integrated firewall. Authentication based on OAUTH2 – with tokens – and password always get parsed through Salted Hash oneway algorithme.

Unity Operator Dashboard Security

Dashboard Environment: This environment allow staff / super users at the operator to manage Unity – product design, rating plans, reports, invoicing and associated.

Dashboard connectivity: This environment are a seperate system that isn’t installed together with the open frontend environment – it is able to interact directly with database content for advanced reporting,invoicing and more complex operations.

Dashboard security: Using same stateless connectivity as frontend, but only staff roles can access this environment and can b limited to only be accessed from internal IP addresses – integrated firewall and login scanner with auto banning.

Unity Backend Environment

Backend Environment: This environment contain all Unity modules for provisioning, billing,monitoring, support and similar.

Backend connectivity: Backend servers are offering Unity API connectivity for Frontend environment, connectivity to Inmarsat, Intelsat, Eutelsat, Iridium, Thuraya and Orbcomm.. for provisioning, CDR file collection as well as SSH tunnel connectivity for remote management.

Backend security: Both nodes have integrated firewall, and have no diect access to Internet except above listed connectivity that is highly limited to the needed connectivity. TLS connectivity with Database Servers - in smaller setup will database be on same server as Backend without TLS.