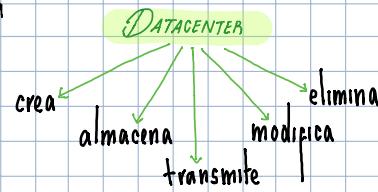


**Espacio Físico**

- sismo resistencia
- piso elevado ~ refuerzos
- ubicación geográfica

**Datacenter**

- \* Generadores \* sistema de respaldo interno.  
+ empresas generadoras se necesitan para subestación.
- \* Transportadores compatibles ~ SNI
- \* Distribuidores de subestaciones a instalación de uso final
- \* Comercializadores
- \* Instalaciones uso final
  - A Fuente Comercial (ONE)
  - B Fuente Autónoma => DIESEL > ALIMENTABLE EN CALIENTE
- \* Hay que tener un transformador independiente
- \* UPS: UNINTERRUPTED POWER SUPPLY: una especie de batería que permite cambiar entre el suministro externo y el mecanismo de generación autónoma sin que haya apagamiento de freq. o tensión.
- NO SE SUPONE COMO SISTEMA DE RESPALDO, SOLO COMO TRANSICIÓN DE ...



**Seguridad:**

### \* Segregación de funciones

Entre las personas, alguien que va a operar, y un garante que sape esta persona que va a hacer (racks, parches, etc.)

### \* Control de acceso

No cualquiera puede entrar, además que no se realiza ensamblaje en el mismo rack sino en un área blanca aparte, y en el rack solo se va a montar todo y salir en poco tiempo.

### \* Protección contra incendios

Hay un gas especia industrial que consume el oxígeno para detener la combustión, pero por esta misma razón a gente debe evacuar por la falta de oxígeno.

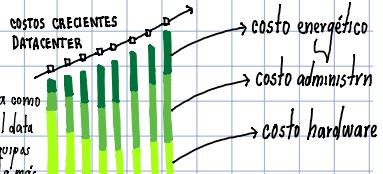
### TRANSFORMACIÓN:

de datacenters físicos a máquinas virtuales, ventajas dinámicas.

NO RENTAR UN RACK QUE PUEDE SER SUBUTILIZADO, SIN INSTANCIAS VIRTUALES COMPARTIDAS EN UN RACK QUE SE ADJUSTRAN DE ACUERDO A LAS NECESIDADES.

### CALOR:

Se pierde energía como calor, se calienta el data center, y algunos equipos no pueden operar a más de cierta temperatura. Además se deben avisos especiales que controlan la humedad, mucha puede causar sublimación y oxidación, y poco causa estatización.



En datacenters baja el costo de hardware, pero sube el costo de administración, energía consumida cooling ...

las transformaciones de energía, cambio de fuentes, y uso no eficiente generan pérdida de calor, es físicamente imposible un 100% de eficiencia, pero se puede llegar muy cerca a este nivel.

**Potencia:**  $V \times I \times FP$

VOLTAJE CORRIENTE

FACTOR POTENCIA

más que todo relacionado a eficiencia, pues toda operación representa una pérdida de energía

En un datacenter, el FP

tiende a ser mayor a 0.9  
osea una eficiencia de 90% +

$$P.U.E = \frac{\text{Watts in}}{\text{Watts ut}}$$

power usage efficiency

idealmente debe acercarse a 1 para evitar pérdidas de potencia

**POTENCIA**

|          |                          |
|----------|--------------------------|
| NOMINAL  | VA $\Rightarrow$ KVA     |
| Efectiva | WATS $\Rightarrow$ KWATS |
| Reactiva | VAR $\Rightarrow$ KVAR   |

pérdida

$$P_N = P_E + P_R$$

USUALMENTE TIENDE A ZERO

SUMA "ESTÁTICA"

**Capacidad de Cómputo**

componentes

\* Hardware de Servidores  
\* Software de virtualización  
\* Sistemas Operativos  
\* Red y Sistemas de Almacenamiento

UNA APLICACIÓN WEB NO REQUIERE TENER UN CLIENTE INSTALADO, LO ÚNICO NECESARIO ES TENER UN BROWSER.

USUARIO  $\rightarrow$  acceso a través de un navegador

Internet  $\rightarrow$  Servidor Web  $\rightarrow$  Servidor de Aplicación

Mainframe  $\rightarrow$  Base de Datos  $\rightarrow$  Backend

SI ES UNA RED INTERNA NO SE NECESITA ACCESO HACIA AFUERA PARA USAR APLICATIVOS INTERNOS

**Servidor**

**CPU:** los servidores se clasifican de acuerdo a su arquitectura de computo

ALU: unidad lógica lógica  
CONTROL: unidad de control  
DATA CACHE  
INSTRUCCION CACHE  
REGISTERS  
DECOD UNIT  
PREFETCH UNIT  
MEMORY BUS

X86 - 64 / Intel / AMD

PowerPC / IBM  
Gama de RISC  
INTEGRITY ~ ITANIUM  
Intel (para HP)  
HP-UX

WINDOWS SERVER

LINUX - CENTOS, RH, SUSE, FEDORA

OPEN SOLARIS

OSX SERVER - descontinuado

LINUX FOR POWER

AS/400 ya la mayoría pre separan los 30/15.

AIX

Sparc / Sun Microsystems  
~ Oracle Hardware

aún vigente hoy en dia

**RAM:** almacena datos e instrucciones que una aplicación necesita procesar

mientras mayor capacidad de memoria y mayor bus, más se puede enviar al procesador. En este momento el cuello de botella tiende a ser la memoria y no el procesador.

**MEMORIA RÁPIDA**  $\rightarrow$  MÁS RÁPIDA, PUES MÁS OPERACIONES / TRABAJA PUP

**INTERFACES:** HBA / HOST BUS ADAPTER

node de alta velocidad para conectarse a almacenamiento de datos.

ETHERNET / REDES LAN

conexiones de red tradicionales para una red LAN o internet.

**Servidor:** un computador físico, dedicado a ofrecer uno o más servicios, sirve las necesidades de usuarios y otros clientes/servidores.

**Procesador:** ejecuta instrucciones, tiene registros para diversas propósitos, con una arquitectura específica entre muchas posibles que causa que varíe su set de instrucciones y que se corre.

**Sistema Operativo:** comunica HW y SW, gestiona los recursos.

**Almacenamiento**

Hard Disk Drive

\* piezas mecánicas

\* bottleneck: RPM

\* form factor:

Solid State Drive

\* solo se mueven electrones

\* se descargan bajo carga intensa

# Factores de Forma

## Torres de servidores

- \* oficinas remotas, 1º pocas máquinas.
- \* ocupan mucho espacio, costos impidiendo.
- \* fáciles a pagar para reparar.
- \* son de bajo costo, pero complejos de administrar.

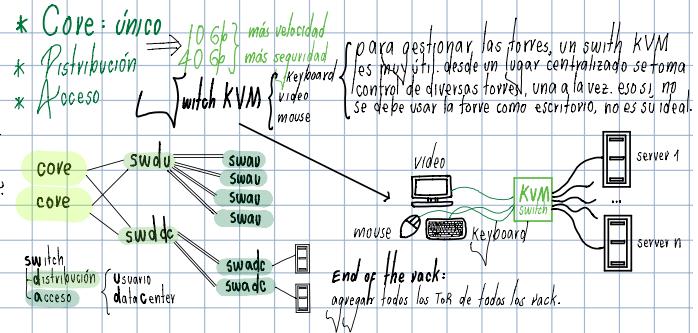
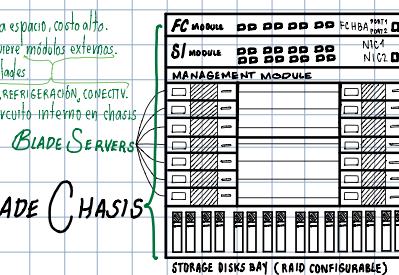
## Servidores de rack

- \* tienen medidas estandarizadas.
- \* bahías de almacenamiento SSD/HDD (SFF/LFF)
- \* puerto gestionado (mejor switch KVM o control KVM móvil).
- \* tienen diversas puertas como SFP
- \* permiten tarjetas de expansión

## Servidores de blade

- \* altamente optimizado en cuanto a espacio, costo alto.
- \* sacrificia funcionalidad interna, requiere módulos externos.
- \* en un chasis se acomodan 8-16 blades.
- \* el chasis incluye: MGMT, POTENCIA, REFRIGERACIÓN, CONECTIVIDAD.
- \* no hay cableado, todo es un circuito interno en chasis.

## REPRESENTACIÓN GRÁFICA BLADE CHASIS



**MGMT:** módulo que facilita la gestión automatizada, monitoreo y controla

**HBA/FC:** módulo para SAN, encargado de la conectividad.

**NIC/ETH:** módulo para LAN, conectividad.

**UPLINK:** conecta switches con TrR

**DOWLINK:** conecta servers con switch

**LACP:** link aggregation protocol / si se danía un uplink poder seguir saliendo

fuentes de poder redundantes, assisted by UPS.

Los datacenters son instalaciones, muy complejas, que constan de bastantes elementos, procesos, personas y regulaciones que se coordinan para apoyar la gestión de la organización. Se debe tener en cuenta, como, proveer energía a estos, la redundancia de fuentes de energía y asecciónes como sucesiones y métodos de respuesta interna con alimentación directa de combustible, dispositivos de transición ininterrumpida, de fuente si otra falla, también considerar medidas de seguridad física y operacional que se deben tener en cuenta como medidas de prevención de incendios, control de acceso, viaje a la red, y la separación de funciones, y finalmente, tener en cuenta cuáles son los requerimientos de software que se piden en hardware con una buena estimación de la capacidad requerida V.S. costos de operación, adquisición, mantenimiento, y beneficio al negocio, para optimizar las reservas.

Uno de los elementos, más clave son los servidores, que cuestan cada vez menos, pero adecuar, administrar, y proyectar el espacio, energía y enfriamiento para su funcionamiento es cada vez más costoso. A la vez hay que tener en cuenta otras características para su adquisición, como su arquitectura, sistema operativo, clase y capacidad de memoria, y si es de torre, rack o blade (en todos los módulos que este último implica), si realmente se necesita adquirir un datacenter como una nube pública, privada, o virtualización de acuerdo a las necesidades que se plasman estimado).



## Cluster de virtualización

Varias máquinas con recursos similares para asegurar disponibilidad.

En virtualización el factor más importante son GHz, los ciclos de reloj pues esta unidad para correr instrucciones

**Máquina Virtual:** software que ejecuta su propio OS, se comporta igualmente a una máquina física, y muchos de sus elementos como interfaces se sustituyen en su equivalente de hardware.

**Hypervisor:** en pocas palabras es un sistema operativo en el cual los recursos gestionados son máquinas virtuales corriendo OSs permitiendo correr OSs encapsulados en VMs, los cuales a su vez son altamente portables, y el aislamiento del hardware brinda independencia de arquitectura.

**Baremetal:** no requiere otro sistema operativo.

**On Top of OS:** corre como una app en un OS

## VIRTUALIZATION DISTRIBUTED SERVICES

HA: cuando falla un host, se transfieren las VM a otras hosts.

FT: se crean 2 VM sincronizadas en dos hosts distintos, si falla uno se continua en el otro, pero hay duplicidad de memoria.

**VMOTION:** migraciones en VIVO

**DISTR. SERVICE HA:** mueve de un host a otro, pero corre una interrupción.

**FT:** crea dos máquinas sincronizadas para la HA.

**VCENTER:** manejo centralizado de las máquinas.

**DISTR. RESOURCE SCHED:** balancea VMs sobre los host.

**DISTR. POWER MGMT:** distribuye según necesidad de humor.

La virtualización del cómputo permite superar previas dificultades como la imposibilidad de correr un sistema operativo de una distinta arquitectura a la del hardware, y la imposibilidad de correr múltiples sistemas operativos paralelamente en un hardware, solucionado con un hypervisor, que permite encapsular una máquina como software en su interior con las adecuadas interfaces de red y otro hardware, y este hypervisor puede ser tanto un sistema operativo que corre sobre el metal tanto como también un aplicativo que corre dentro de un sistema operativo normal. Los recursos entre estas máquinas virtuales son independientes, pero se acceden por demanda, es posible establecer un límite, y un fallo en una de las máquinas no representa un fallo de la máquina en general. Las máquinas están aisladas, pero siempre existe la posibilidad que una falla de muy bajo nivel permita escapar el hypervisor. Lo más importante son los ciclos de reloj de la máquina subyacente, pues ésta es la unidad de cantidad de instrucciones para correr. Para la virtualización, se pueden implementar clusters que permiten asegurar disponibilidad y hacer eficiente la asignación de recursos mediante el uso de schedulers y control centralizado. La mayor ventaja de las máquinas virtuales es su portabilidad, gracias a esto es posible moverla entre máquinas físicas en caso de falla, o incluso cargar en una máquina física una máquina virtual ya provisionada con ciertas características, permitiendo fácilmente tener varias máquinas casi homogéneas a partir de máquinas físicas posiblemente heterogéneas para incrementar la compatibilidad. En los escenarios de pasal de máquinas físicas a virtuales, hay que tener en cuenta la capacidad utilizada y no la asignada, y multiplicar el Número de cores \* velocidad \* índice utilización (que se obtiene de herramientas de monitores de CPU, % de uso). El % no puede ser el promedio de todos los datos, sino el de los picos máximos registrados. A la vez, hay que permitir una holgura que permita variaciones pero excluya los picos atípicos, y a la vez debe ser la información de las horas de negocio donde se registra la actividad real. Luego se debe estimar el crecimiento, y buscar productos que maximizando el producto de número de cores y la velocidad. A la disponibilidad, que pasa si se cae uno, serán necesarios meter otra máquina o hacer uso de la holgura.

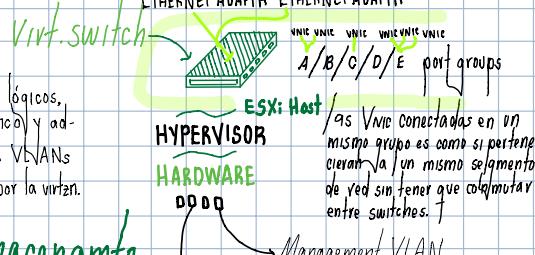
## Módulos de Interface de Red

Flex Ethernet FCoE (encapsulado sobre ethernet)

FlexFabric Ethernet a Fiber Channel (puro)

Flex NIC y /a virtualización

divide un puerto físico en cuatro puertos lógicos, repartiendo ancho de banda, monitorizando y administrando el ancho de banda de distintas VLANs dependiendo a la competencia por recursos traída por la virtual.



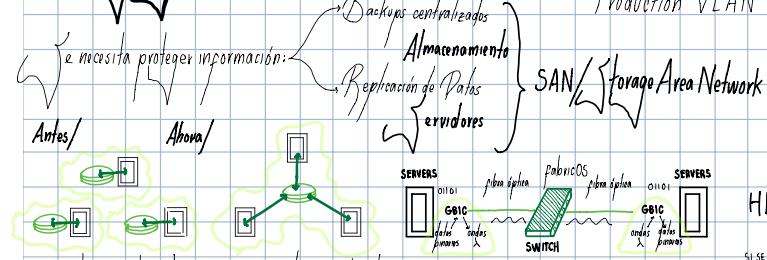
## Scheduler ~ CoScheduling

para n-cores de una VM, el algoritmo de cosched. tiene que encontrar n-cores físicos con un ciclo de reloj libre en el cual ubicar los n-cores virtuales. en el mismo instante de tiempo.

+ cores en una máquina virtual

- probabilidad encontrar cores físicos disponibles (en cosched- utilizando los cores lógicos HT no cuentan)

## Redes y Sistemas de Almacenamiento



SCSI: protocolo nativo para mover datos, encapsulado eth.

HDD: discos duros magnéticos, con piezas móviles e imanes

SSD: discos de estado sólido, solo se mueven en electrones

HDD / 7,2 RPM  
SSD / 10000 RPM  
Se guardan a las revoluciones dependiendo el 10PS.  
Si se necesitan X TB EN BASE 2  
TiB Tipos 5,7 TiB  
= 10<sup>12</sup> TERAS EN PUBLICIDAD  
= 2<sup>32</sup> T<sub>2</sub>  
TERAS VISTAS POR D.S.  
= 6.12 T<sub>10</sub> TERAS PUBLICITADAS

Un sistema de almacenamiento central provee tolerancia a fallos, escalabilidad, etc

### Raid 0



Reporte la información. Es un disco efectivamente de tamaño doble, facilita escribir en ambos al tiempo, pero si muere uno se pierden todos.

$$N * X$$

### Raid 1



suma desempeño en lectura (no completa en ambos a la vez), más lento para escribir (debe escribir en ambos) no se duplica la capacidad.

$$X$$

### Raid 5



Mínimo tres discos, hasta 16. Se busca usar discos de mayor tamaño con algunos datos provenientes de otros discos 2 por si falla cualquiera de los 2 (se pierden).

$$(1 - \frac{1}{n}) * (N * X)$$

### Raid 6



Es como el raid 5, pero con mínimo cuatro discos, para que cualquiera dos discos puedan fallar.

$$(1 - \frac{2}{n}) * (N * X)$$

### Raid 10 1+0



Es raid 0 y 1 al mismo tiempo de manera que se realiza striping y mirroring a la vez, el único inconveniente es la cantidad de discos.

$$2 * N * X$$

La infraestructura cloud cuenta con diversos bloques de construcción, como lo son los data centers, el cómputo, el almacenamiento, la colectividad y el software de cloud. Además este software de cloud se encarga de la orquestación, la automatización de procesos, y el frontend de usuarios, también está la plataforma cloud y el software cloud, donde cada nivel es una agregación del previo. La infraestructura se puede construir de distintas formas, todas con los mismos building blocks pero separados de una forma distinta. Una arquitectura de referencia es una receta de un fabricante donde se dan unos lineamientos de estructuración. Donde se prefiere que todo sea de un mismo fabricante, pero se puede usar de otros pues todo es inter-operable gracias a estándares, aun así toca revisar bien cada update que no vaya a romper la compatibilidad entre distintos dispositivos. La arquitectura convergente es una solución empaquetada de diversos componentes. Con una capa de gestión común, donde se sacan updates para todo en un solo paquete, es muy costosa, y lo único que es separado es el storage centralizado con un típico acceso de fiber channel. En la arquitectura hipervconvergente incluso el storage esta incorporado en el paquete, no se usan blades ni racks sino appliances especializados, desaparecen las SAN pero aparecen LAN todo orquestado por software y comunicaciones a través de ethernet, útiles para pequeñas empresas que desean crecer rápido, bastante costosos pero el tamaño y capacidad por módulo representa un costo bajo si se necesita rentar espacio con un bajo consumo de energía.

# Cloud Computing

permite el acceso desde cualquier lugar, de manera fácil y por demanda, a recursos de computación compartidos y concub.

**Autoservicio** se realiza por demanda

Acceso por **tipos req** privilegiando IP/ETH

Recursos Compartidos publico: contratar / priv: con servicios en == org.

Rápida Elastisidad

Medición de Servicio fácil medir costos asociados a distintos recursos.

## \* Infraestructura Cloud

- \* Datacenter
- \* Orquestar
- \* Automatizar
- \* Front Usuarios
- \* Cómputo
- \* Almacenamiento
- \* Conectividad
- \* Software de Cloud

## \* Plataforma Software

para el storage también hay virtua & hypervisores, pero es muy costoso.

## modelos servicio

**IAAS** proporcionar recursos fundamentales de cómputo  
INFRASTRUCTURA SISTEMAS OPERATIVOS, STORAGE, RED.

**PAAS** crear y desplegar apps  
PLATFORM APPS DESPLEGADAS Y ENTORNO RUN.

**SaaS** ofrecer software  
SOFTWARE EL SOFTWARE MISMO ES EL SERVICIO

## modelos impl.

**PRIVADA** \* utilizada por una org.  
\* propias inst. o de un 3ro.

**PÚBLICA** \* compartidas por varias org.  
\* esta en las inst. del proveedor.

**HIBRIDA** \* combinación public y priv.  
\* el futuro!  
\* despliegue para **propósitos**

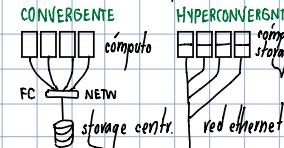
**PRIVADA:** PRODUCCION **PÚBLICO:** DEV, TOL, FALLO

se puede construir con

### \* ARQUITECTURA REFERENCIA

### \* ARQUITECTURA CONVERGENTE

### \* ARQUITECTURA HYPERCONVERGENTE



una "receta" de un fabricante, todos son inter operables por standards, pero menor == 2º fabricante. Toda revisar compatibilidad de equipos. Todo lo demás.

una única máquina empalmeada de diversos componentes con una capa de gestión común. Un solo update trae todo con todos los discos. Solo está separado el almacenamiento (centralizado) más flexible, dinámica escalable que la anterior. basada en bloques funcionales (servidores) que conectan por ethernet, no hay una SAN física, sino SAN virtuales. El storage y los computo viven juntos, no son blades ni rack sino aplicaciones especializadas. Utíl para empresas pequeñas que buscan crecer rápido.



File I/O  
Block I/O

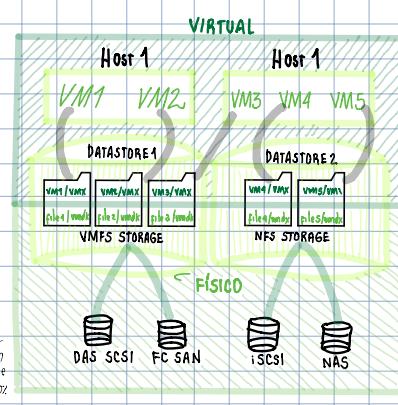
Filesystem: una partición

- \* Tiene una lógica
- \* Tiene unos limitadores

para escribir en un disco se necesita conocer su ubicación y la forma de escribir

Replicas Sincrónicas ~ baja latencia

...  
Syncrónicas  
Se necesita un buen medio, los softwares de replicación necesitan de alta convergencia para que uno transcriba se responda al otro con alta velocidad



LUN ~ LOGICAL UNIT NUMBER

Identificador único para colecciones de almacenamiento físico o virtual

Un filesystem es una partición con unos delimitadores de ubicación y una lógica con la cual se le escribe. Para escribir en un disco y leer de él se necesita conocer estas de limitaciones y su lógica o formato. En una arquitectura de almacenamiento se divide la parte virtual y la física. En los sistemas de almacenamiento están los procesadores, luego los controladores RAID luego los discos físicos y las interfaces de red HBA o FC. Los hosts con las VM corriendo se conectan a los data stores que contienen los folder de VM, ya sea a un storage NFS o VMFS especializado, que ya transición a de lo virtual a lo físico mediante conexiones FC o eternos a DAS SCSI FC SAN iSCSI o NAS. Los LUNS son segmentaciones de disco. El I/O se da en File o Block, Block I/O es cuando se determina el FS de acuerdo a la aplicaciones, usado en hypervisor o almacenamiento, cuando la App determina cuál puede ser el mejor, mientras que el File I/O se da cuando el Sistema Operativo conoce el FS y no lo determinan las apps. FC SAN, más costo por involucrar HBA, pero mayor velocidad por uso de fibras, comparado con el cobre en network attached, pero menor costo al usar LAN tradicionales. Los métodos SAN como fiber channel o a través de ip permiten usar file io o Block io, mientras que los métodos attached como direct attached y network attached solo permiten file io, el directo mediante puertos pci mientras que network attached necesita de NICs o switches LAN. Las réplicas sincrónicas son necesarias para mantener una transmisión de la información rápida para la consistencia entre sistemas, un mayor nivel de convergencia puede representar costos muy altos, lo mejor para esto es usar fibras monomodo.

**Fibre SAN:** FIBER CHANNEL  
\* Fibra  
\* Fiber Channel  
\* HBA  
\* Switches SAN  
\* File I/O, Block I/O

**IPSAN:** IP SAN  
\* Cobre/Fibra  
\* iSCSI, FCoE  
\* NIOs  
\* Switches LAN  
\* File I/O, Block I/O

**DAS:** DIRECT ATTACHED STORAGE  
\* PCI  
\* SCSI  
\* File I/O

**NAC:** NETW. ATTACHED  
\* Cobre  
\* FCoE  
\* NIC  
\* Switches LAN  
\* File I/O

**File I/O** el sistema operativo sabe como está estructurado, archivos normales.

**Block I/O** depende de las aplicaciones, común en hipervisors bases de datos, etc.

**IOPS** input y outputs por segundo depende de los RPM a la latencia promedio, y el tiempo de búsqueda promedio.

# A Two SPEED IT ARCHITECTURE For The DIGITAL ENTERPRISE

## NEW DIGITAL ARCHITECTURE

need to run alongside

## OLD LEGACY SYSTEMS

\* Focus on user friendliness & speed, agility, constant changes

need to iterate fast to keep up with changing customer needs & technologies

\* focus on stability, availability and extensive testing

### PHYSICAL USER CHANNELS

retail, ATM, phone support, etc.

need to occur in parallel, not sequentially

### DIGITAL USER CHANNELS

apps, websites, etc.

digital & physical channels now need to be fully seamless & transform insights

### CROSS CHANNEL DEPLOYMENT

Design services so that they can be accessed from all channels and transition between them in a seamless, supported, digital, integrated, connecting, tailoring & personal way across multi. channels.

information from users allows to inform insights about behavior which supported products, integrating tailoring & personal across multi. channels.

not stand alone, its integral part of operational processes

### BUSINESS PROCESSES

can give insights & transform

big data

### MICROSERVICES

many microservices can come and go, built around core processes, but with a modular nature that allows to keep functionality separated and using many different tools & languages exposing the benefit of each

### REAL TIME ANALYTICS

The ability to analyze all of an user's actions when they are using an app is key to know how to identify and understand all insights and actions.

with microservices this could be done by just deploying more instances of a service if its needed & load balancing load platform.

### AUTOMATED SCALING

work loads may expand & become harder to predict, load balancing across public/profiting environments and fail back mechanism prevent outages, so that others can take load.

\* CYBERSECURITY must be internal, company may have a lot of valuable, sensitive data, and services may act as entry points for hackers (to exploit vulnerabilities). Also, service availability can be disrupted as a result of such malicious attacks.

BUILDING BLOCKS OF DIGITAL-ENTERPRISE ARCHITECTURE

### PRODUCT FACTORY

Also the processes can be designed from products, so that many products can be produced & demand can be constantly changing without having to transform the core processes and the systems beneath them.

### RUN WAVES OF CHANGE AS THREE PARALLEL RUNNING STREAMS

- \* DIGITAL TRANSFORMATION
- \* SHORT - TERM OPTIMIZATION
- \* ARCHITECTURE TRANSFORMAION

### MANAGE HYBRID ARCHITECTURE WITH VERY DIFFERENT PLATFORMS

target architectures are heterogeneous, with system optimized for different environments and critical systems of must be kept down to a minimum, being slowed down by components of old/new HAN.

### PLAN ONGOING SOFTWARE DELIVERIES BY BLENDING METHODOLOGIES

transit can't be done in waterfall or long drawn phases, but also don't change anything to Agile, instead it should fall into the domain of incremental delivery. So there is delivery (customer opps), slow carbon delivery (feature teams)

### DEVELOP THE LOW SPEED ARCH TOO

from the beginning need to establish both capabilities, and then walk away, the low speed arch is very important because transactional batches requires long development, maintenance for resilience/scalability & compromise.

### BUILD NEW ORGANIZATION & GOVANCE MODEL IN PARALLEL WITH NEW TECH

the live business & IT area stands to play, it is no longer just support but its central to the business ops. change everyone's mindsets

### CHANGE EVERYONE'S MINDSETS

since it is now a very critical component of business, having a key role in a company's competitive business, the digital transformation requires more attention and a place on the priority agenda, and need to consider spent money not as a necessity, but a priority, an investment.