

Resiliencia en Sistemas Computacionales Complejos

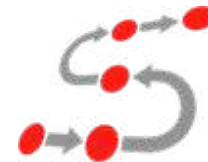
Pedro R. D'Argenio

Dependable Systems Group
FaMAF, UNC - CONICET

<http://dsg.famaf.unc.edu.ar>



Conferencia Gaviola, Noviembre 2015



Resiliencia en Sistemas Computacionales Complejos

Pedro R. D'Argenio

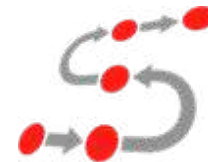
Grupo de Sistemas Confiables

FaMAF, UNC - CONICET

<http://dsg.famaf.unc.edu.ar>



Conferencia Gaviola, Noviembre 2015

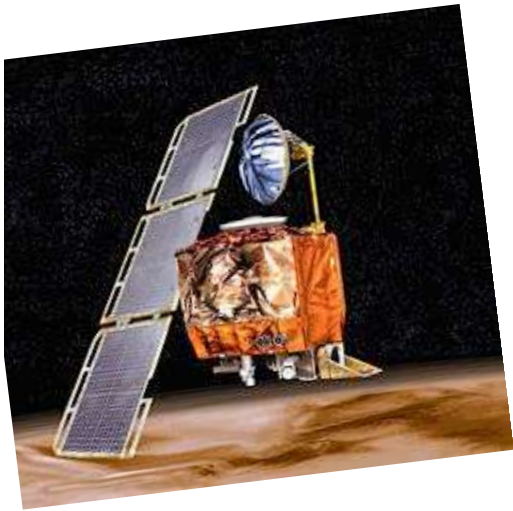


El problema de la corrección

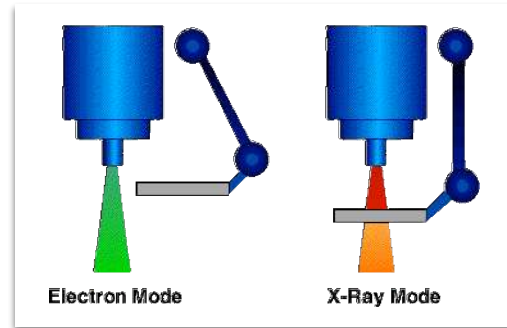


Pentium:
FDIV

Ariane 5:
64 bits fp
vs 16 bits int

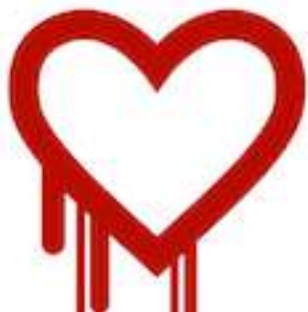


Mars Climate
Orbiter:
Métrico vs Imperial



Therac-25:
Condición de
carrera

Northeast blackout
en 2003:
Condición de carrera



Heartbleed:
Integridad/Confidencialidad

El problema de la corrección

Propiedad \models Sistema

Describe lo que se
espera del sistema
(el criterio de corrección)

Usualmente una
abstracción que describe su
comportamiento

El problema de la corrección

Propiedad \models Sistema

Describe lo que se
espera del sistema
(el criterio de corrección)

Usualmente una
abstracción que describe su
comportamiento

Ejemplos:

- La palabra retornada es igual a la solicitada

Asercional:

```
{n = N & p = palabra}  
  r := heartbeat(n,p)  
{r = palabra}
```

El problema de la corrección

Propiedad \models Sistema

Describe lo que se espera del sistema
(el criterio de corrección)

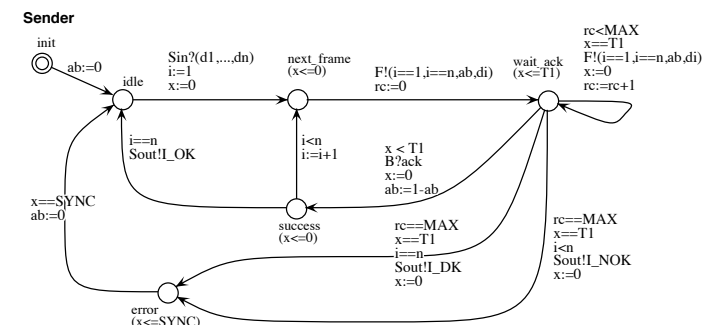
Usualmente una abstracción que describe su comportamiento

Ejemplos:

- La palabra retornada es igual a la solicitada
- El haz de electrones de alta potencia solo funciona bajo la protección del filtro difusor

Model Checking:

$\square ((haz = on) \Rightarrow (filtro = activo))$



El problema de la corrección

Propiedad \equiv Sistema

Describe lo que se espera del sistema (el criterio de corrección)

Usualmente una abstracción que describe su comportamiento

Ejemplos:

- La palabra retornada es igual a la solicitada
- El haz de electrones de alta potencia solo funciona bajo la protección del filtro difusor
- Los datos de la trayectoria leído se corresponden con los reales

Testing y simulación:

The screenshot shows a Python IDE interface. On the left, there is a 'Namespace' window with a table of variables:

Name	Type	Repr
get_version	function	<function get_version at 0x1ef45e0>
image_from_b	function	<function image_from_base64 at 0x1e0a5f0>
keyword	module	<module 'keyword' from '/usr/lib64/python2.5/k...>
listmix	module	<module 'wx.lib.mixins.listctrl' from '/usr/lib64/...>
main	function	<function main at 0x1ef4660>
__class__	type	<type 'function'>
__doc__	NoneType	None
__name__	str	'main'
func_closure	NoneType	None
func_code	code	<code object main at 0x18025d0, file "/data/sys...>
func_default	NoneType	None
func_dict	dict	{}
func_doc	NoneType	None
func_globals	dict	{'DLG_EXPR_TITLE': 'Enter Expression', 'LICENSE'
func_name	str	'main'
open_new	function	<function open_new at 0x1ee578>
os	module	<module 'os' from '/usr/lib64/python2.5/os.pyc'>
pickle	module	<module 'pickle' from '/usr/lib64/python2.5/pick'>
re	module	<module 're' from '/usr/lib64/python2.5/re.pyc'>
rpdb2	module	<module 'rpdb2' from '/data/sys/bin/windbg-1.3...>

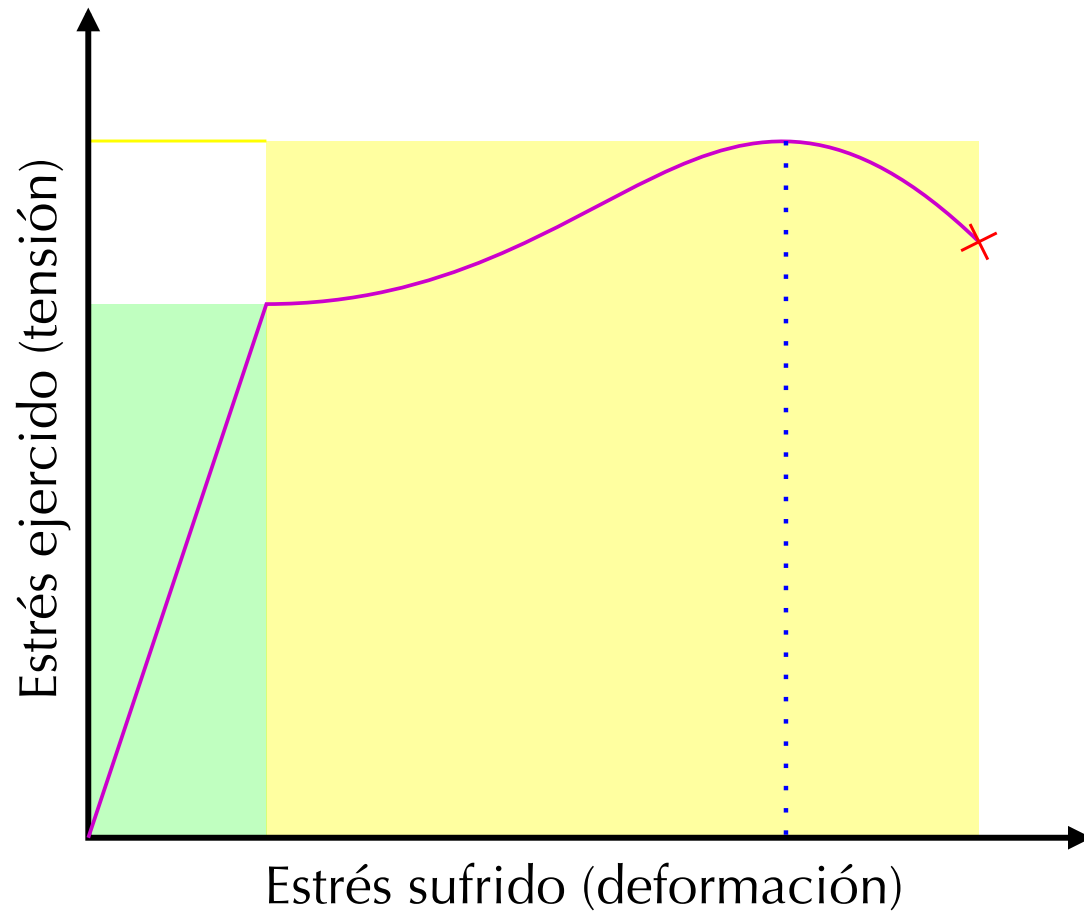
On the right, the 'Source' window shows the code for `main()` in `rpdb2.py`:

```
1522 def main():
1523     if rpdb2.get_version() != "RPDB_2.1.6":
1524         rpdb2._print(SUB_ERROR_INTERFACE_COMMENTFILE
1525         return
1526     return rpdb2.main(StartCtrlUnit)
1527
1528
1529 def get_version():
1530     return WINDBG_VERSION
1531
1532
1533 if __name__ == '__main__':
1534     main()
1535
1536
1537 # Debugging breaks (press) here
1538 # before program termination.
1539 #
1540 # You can step to debug any exit handlers.
1541 #
1542 rpdb2.setbreak(0)
1543
1544
```

Sin embargo...

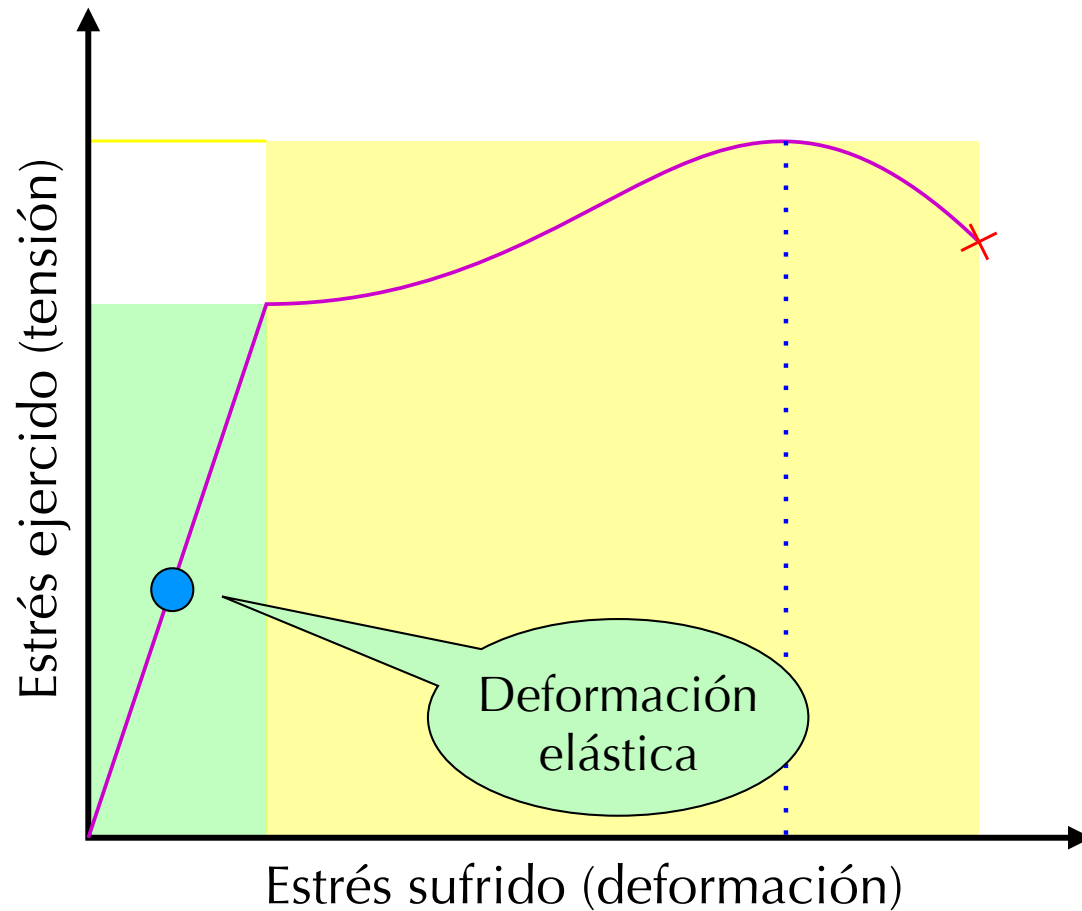


Resiliencia



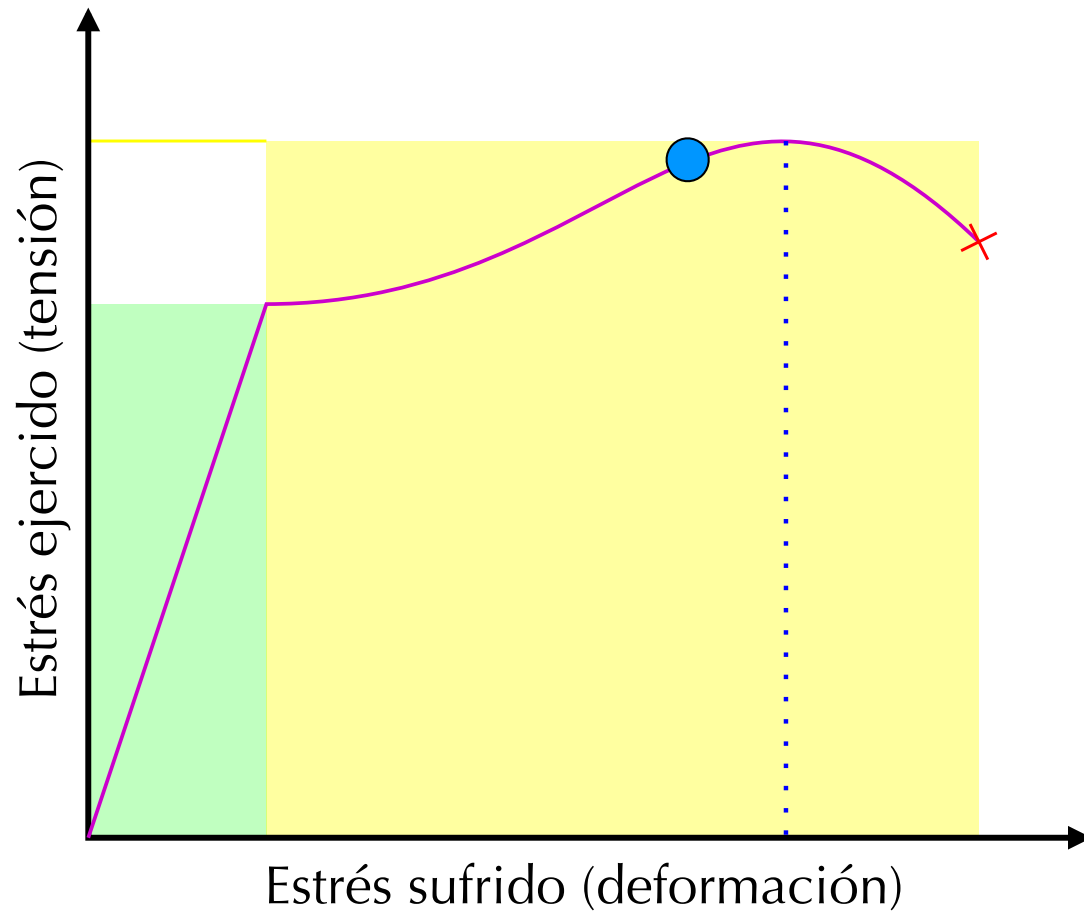
Curva resultante de un ensayo de tracción en algún material (ej: acero)

Resiliencia



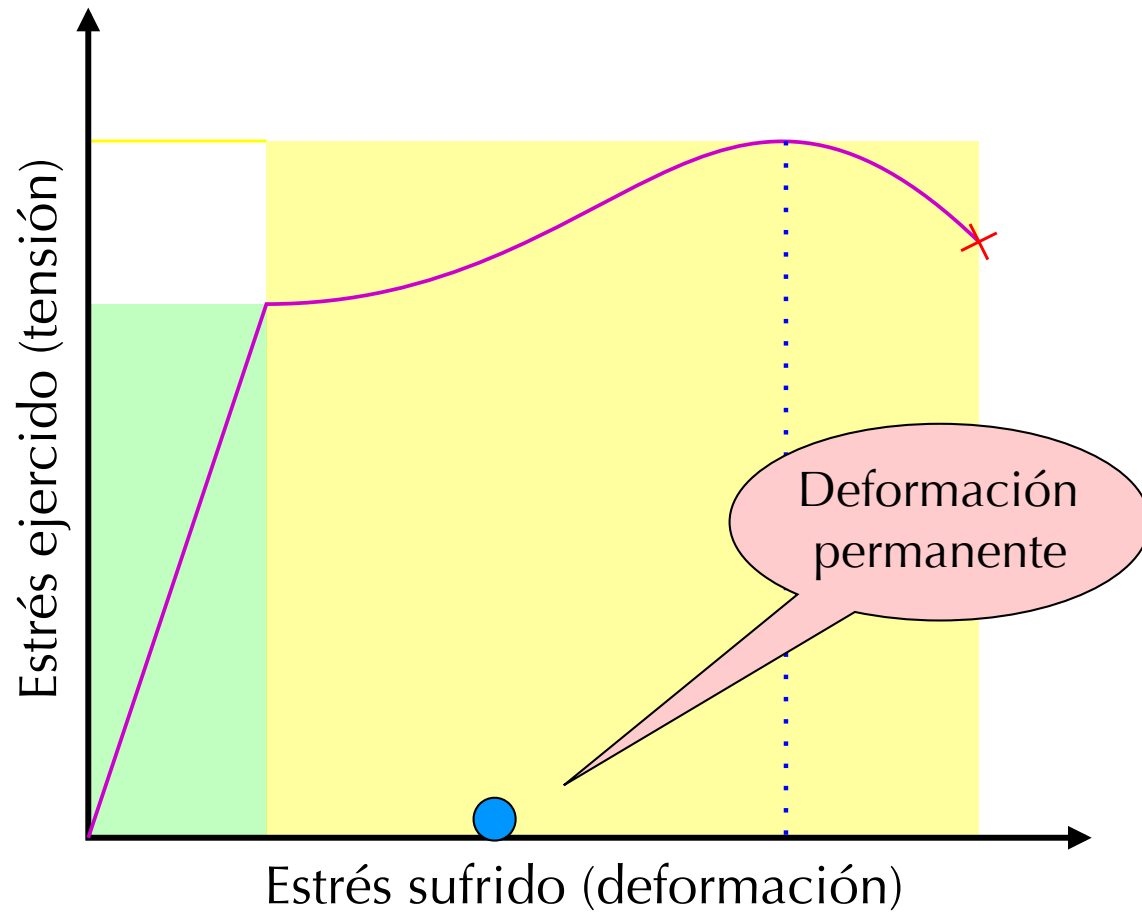
Curva resultante de un ensayo de tracción en algún material (ej: acero)

Resiliencia



Curva resultante de un ensayo de tracción en algún material (ej: acero)

Resiliencia



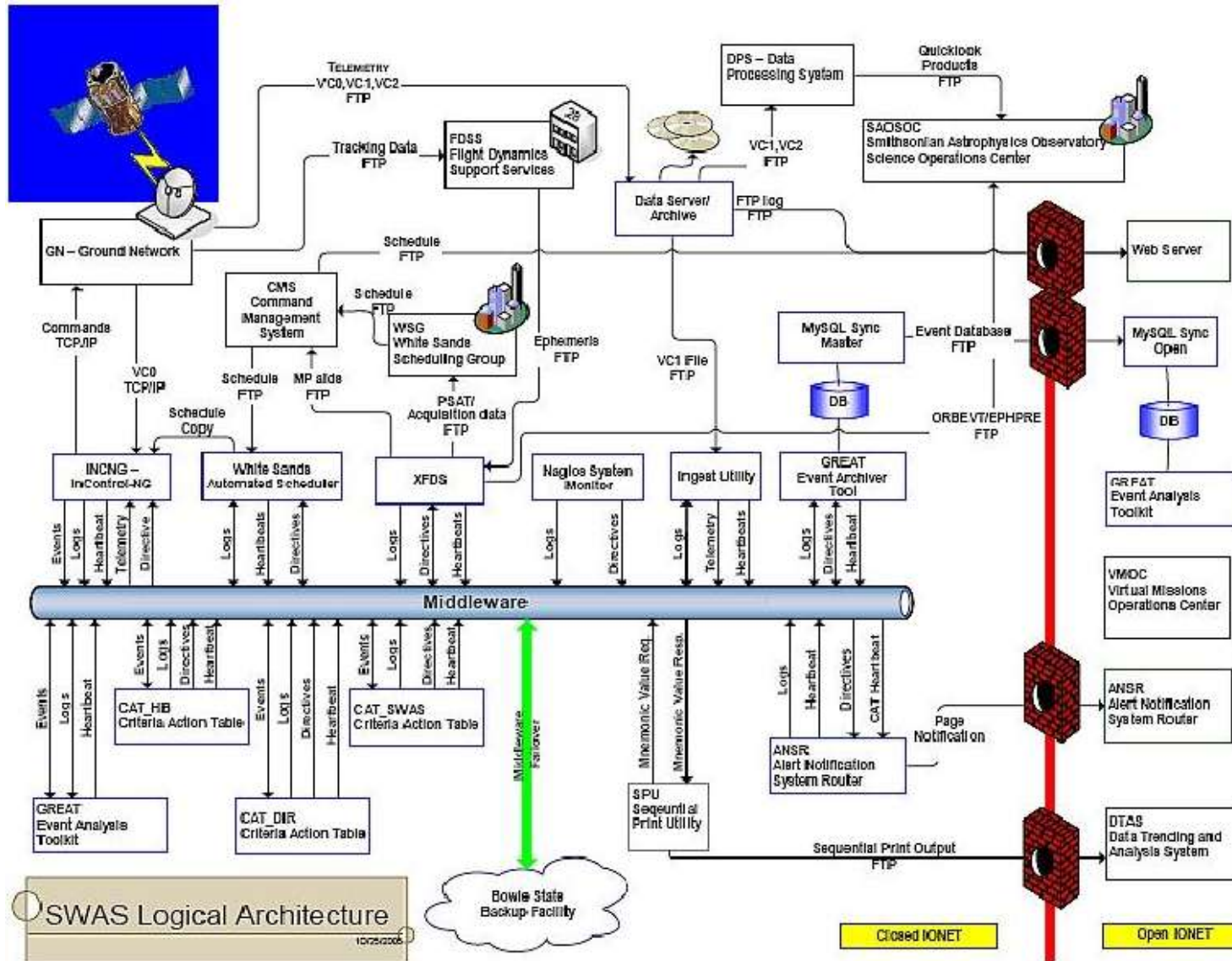
Curva resultante de un ensayo de tracción en algún material (ej: acero)

Resiliencia

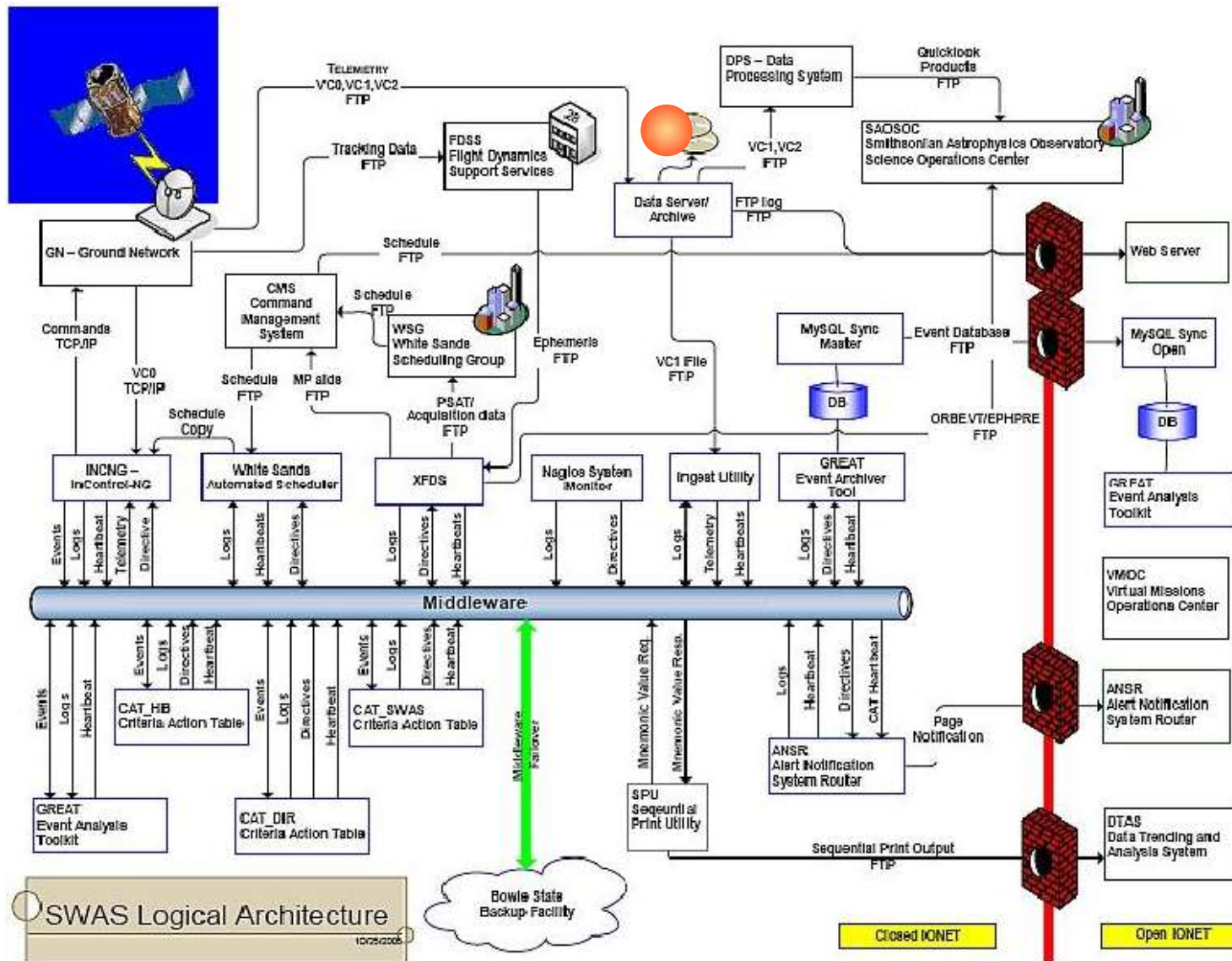
(en Sistemas Computacionales)

La habilidad de **proveer y mantener un nivel de servicio aceptable** aún **bajo la presencia de fallas** y otros inconvenientes que puedan surgir y presentar un desafío al funcionamiento normal del sistema.

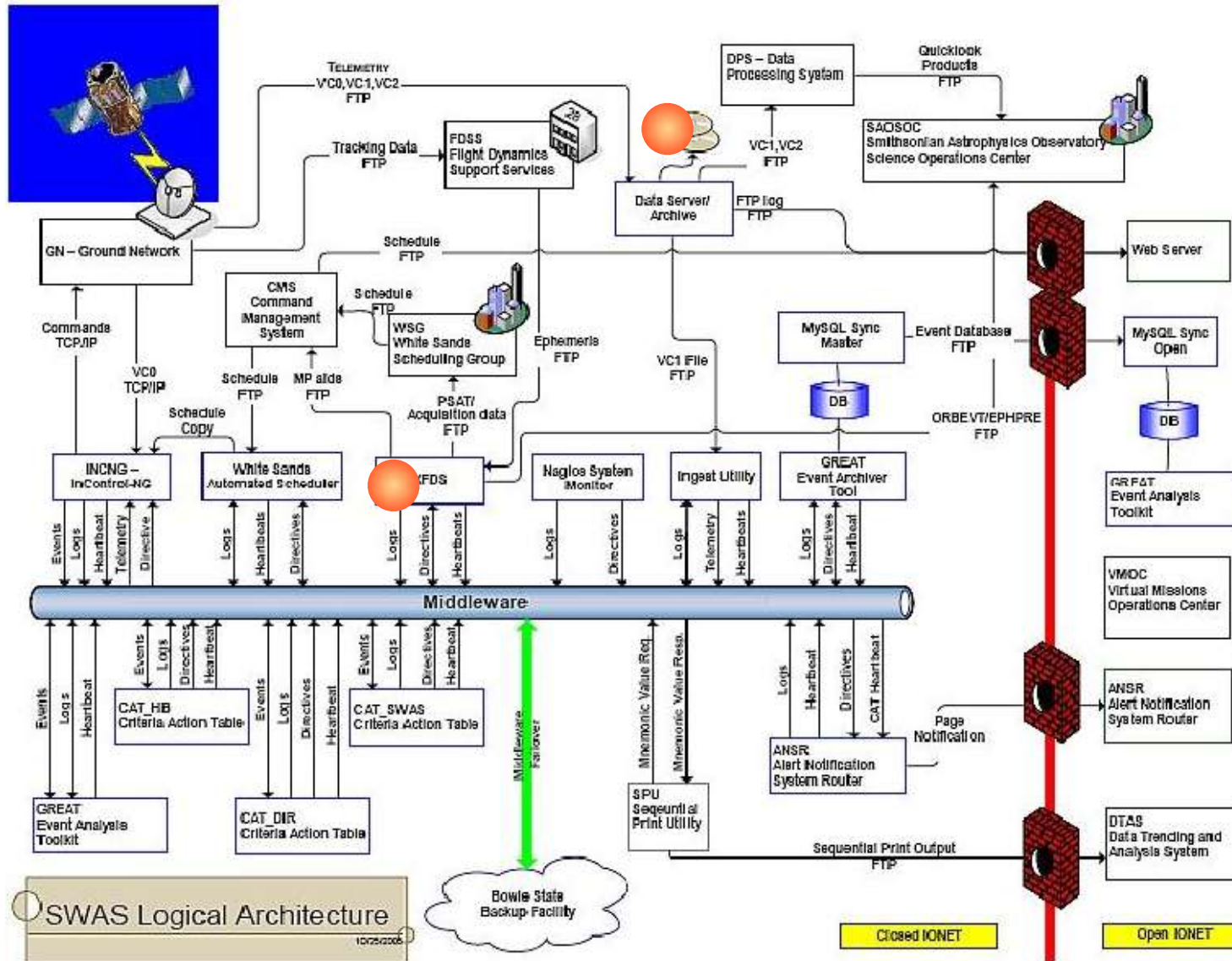
Resiliencia



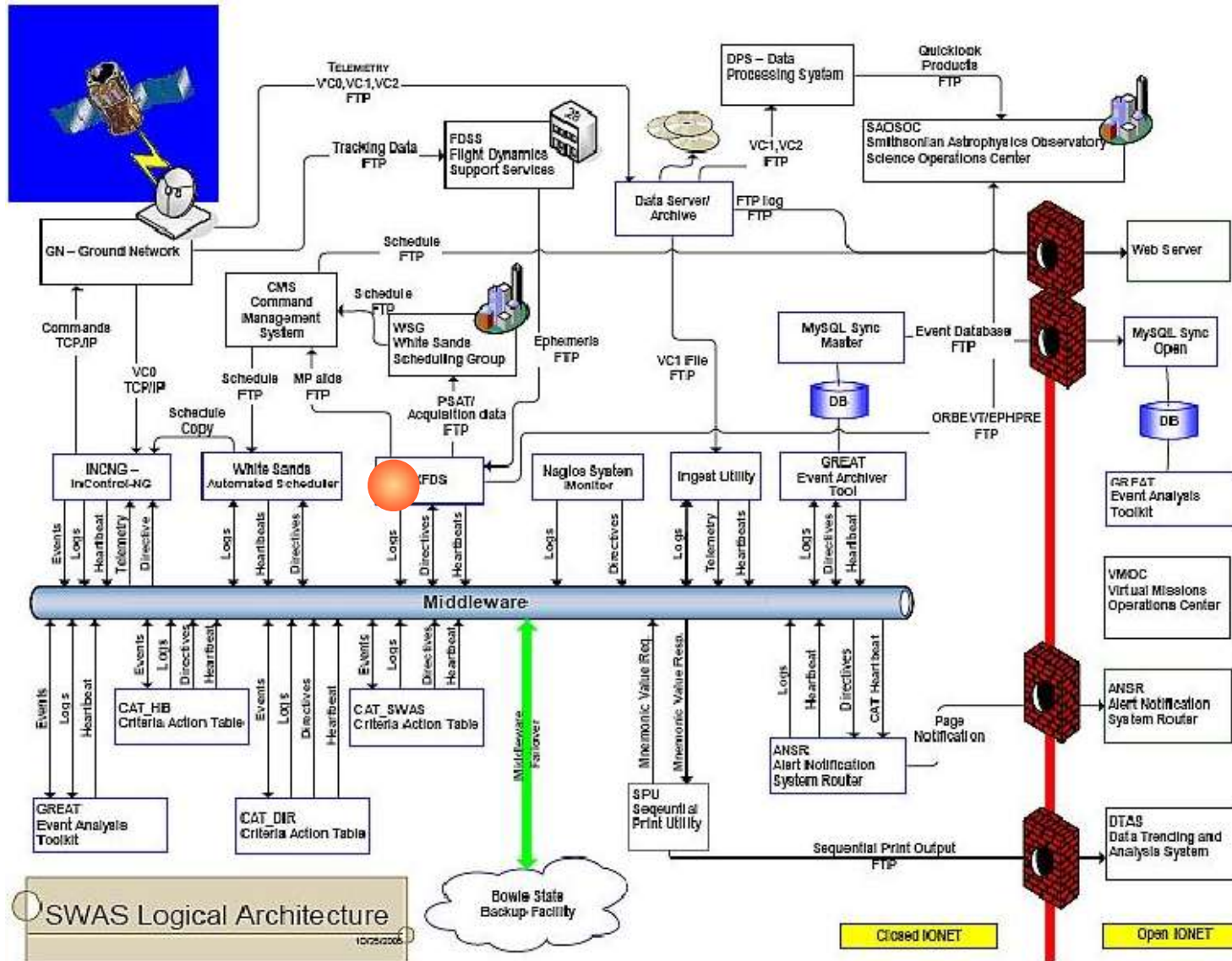
Resiliencia



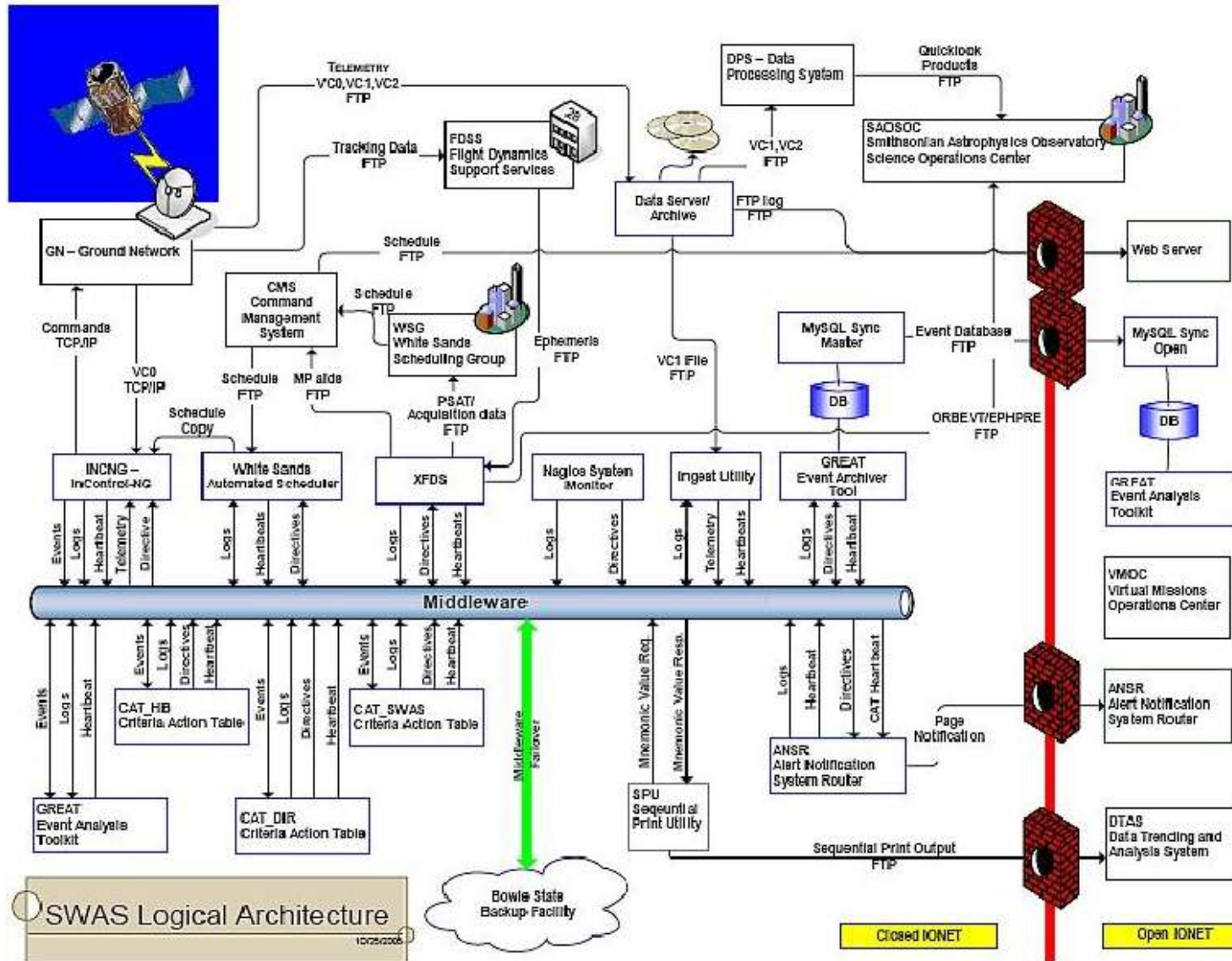
Resiliencia



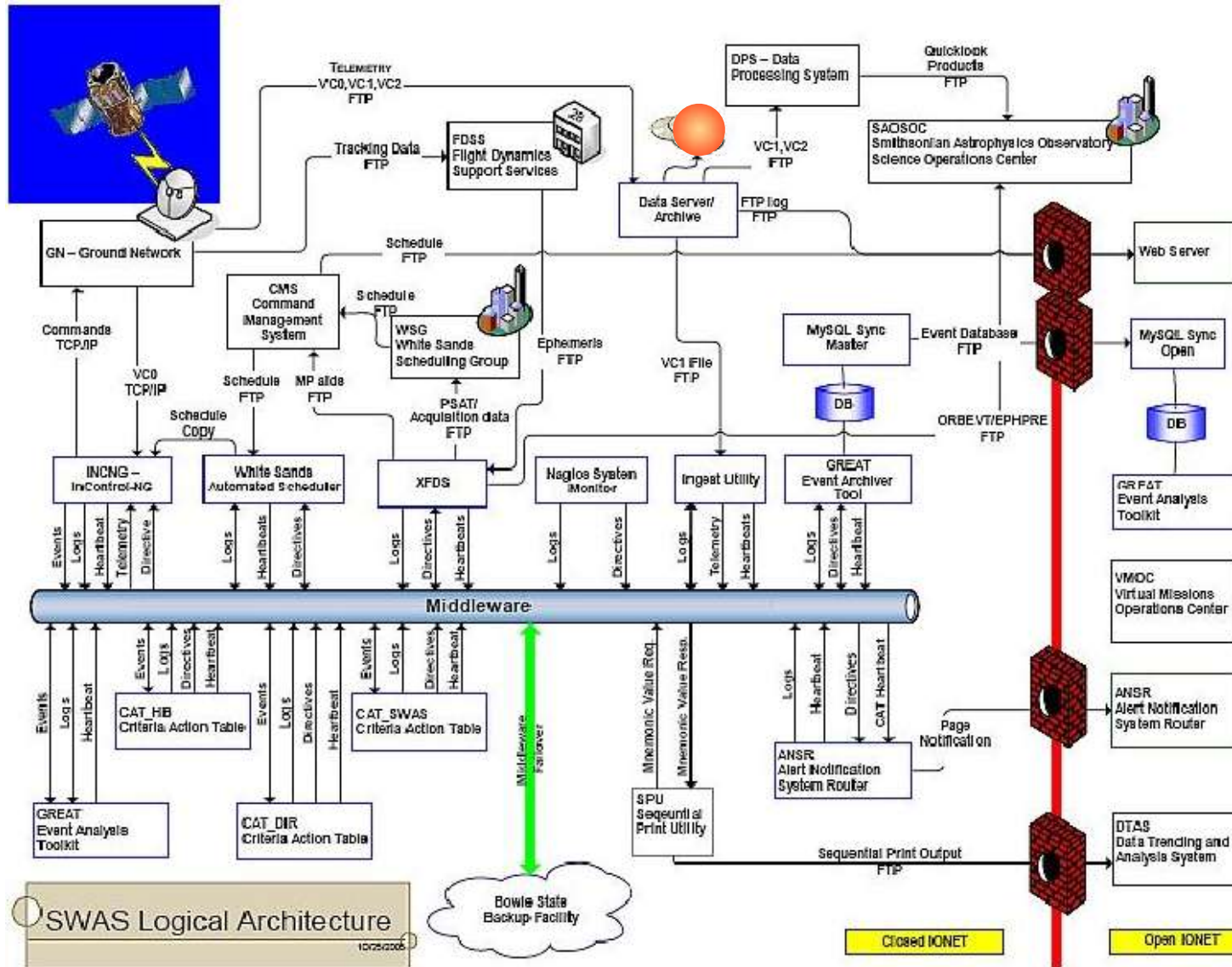
Resiliencia



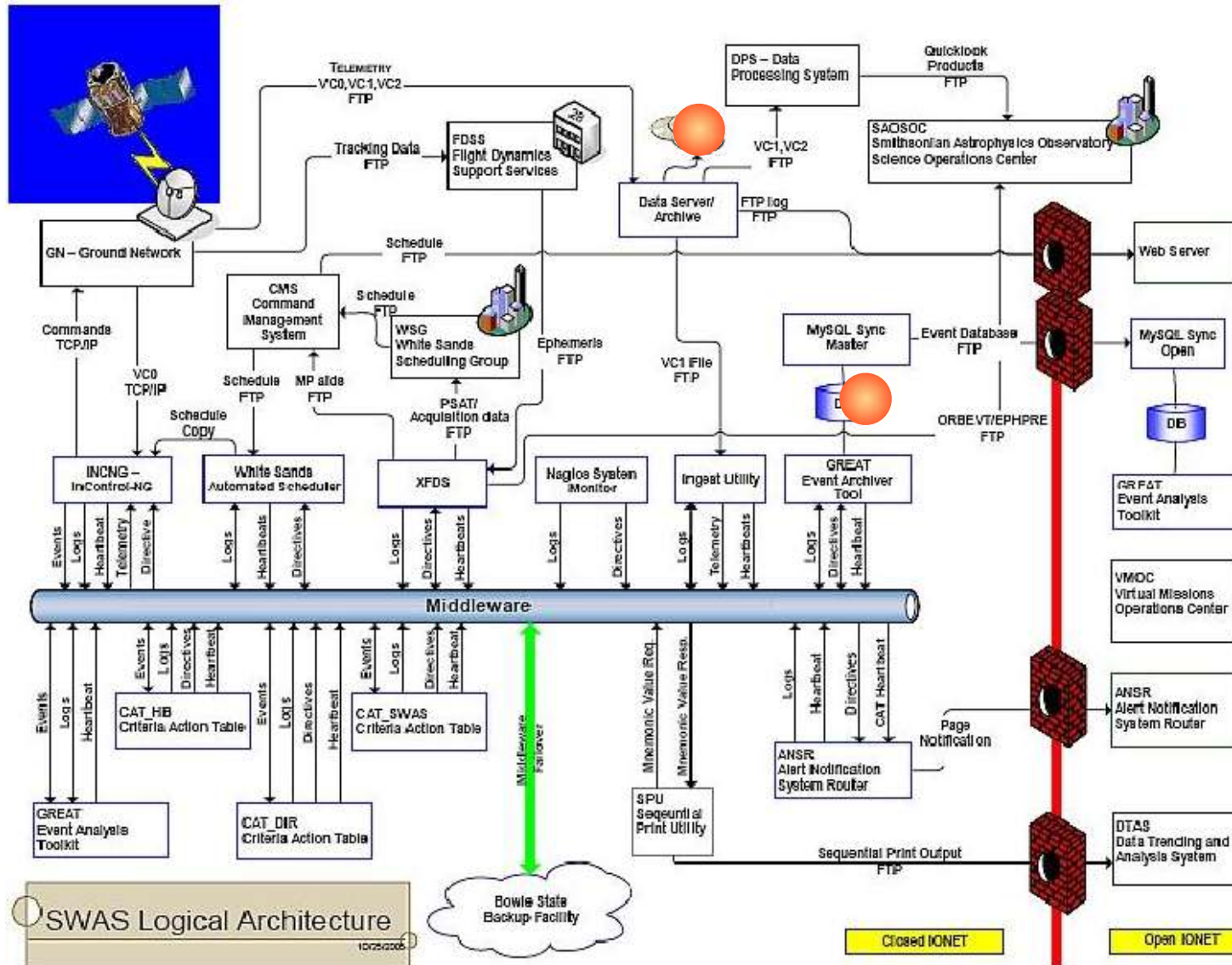
Resiliencia



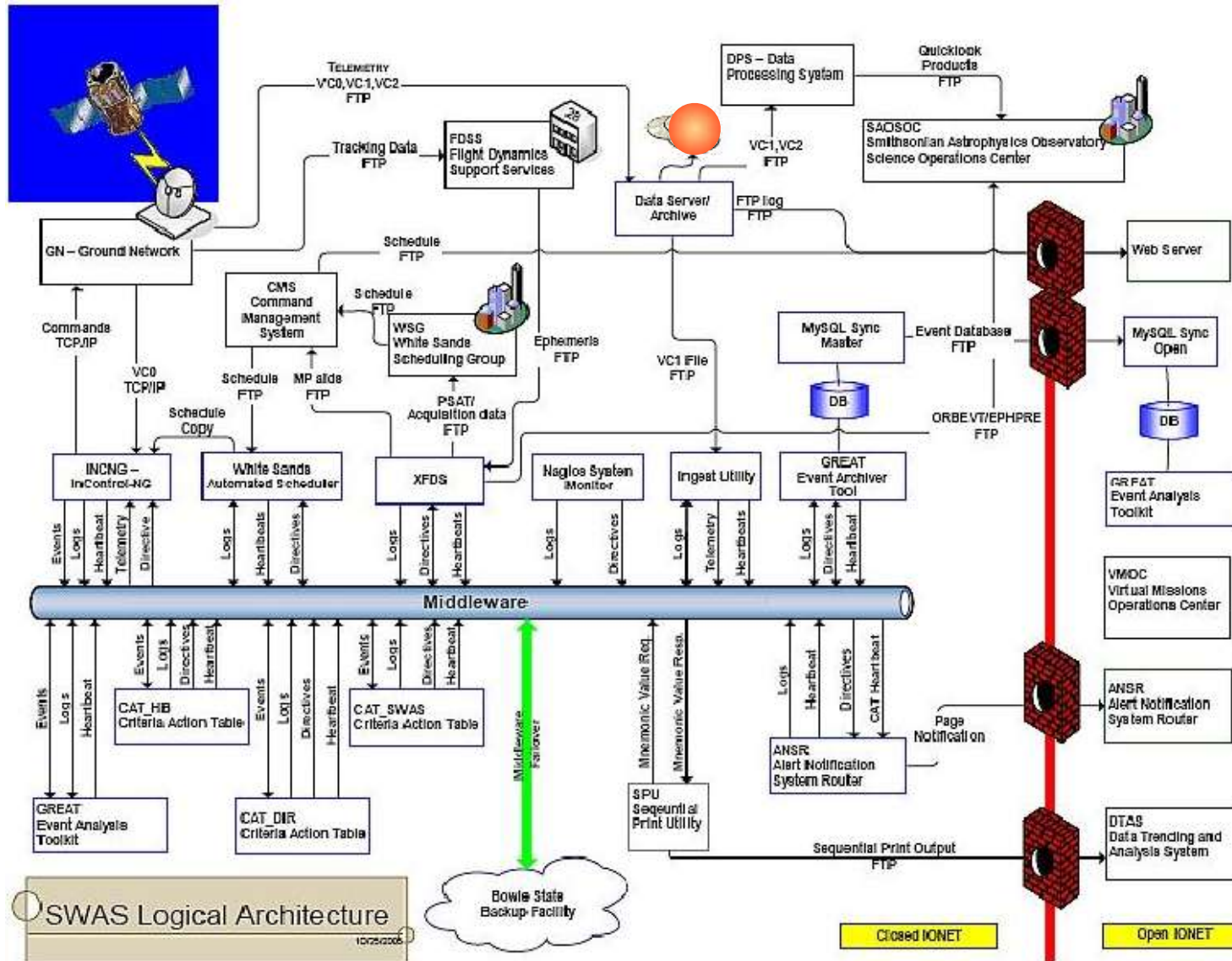
Resiliencia



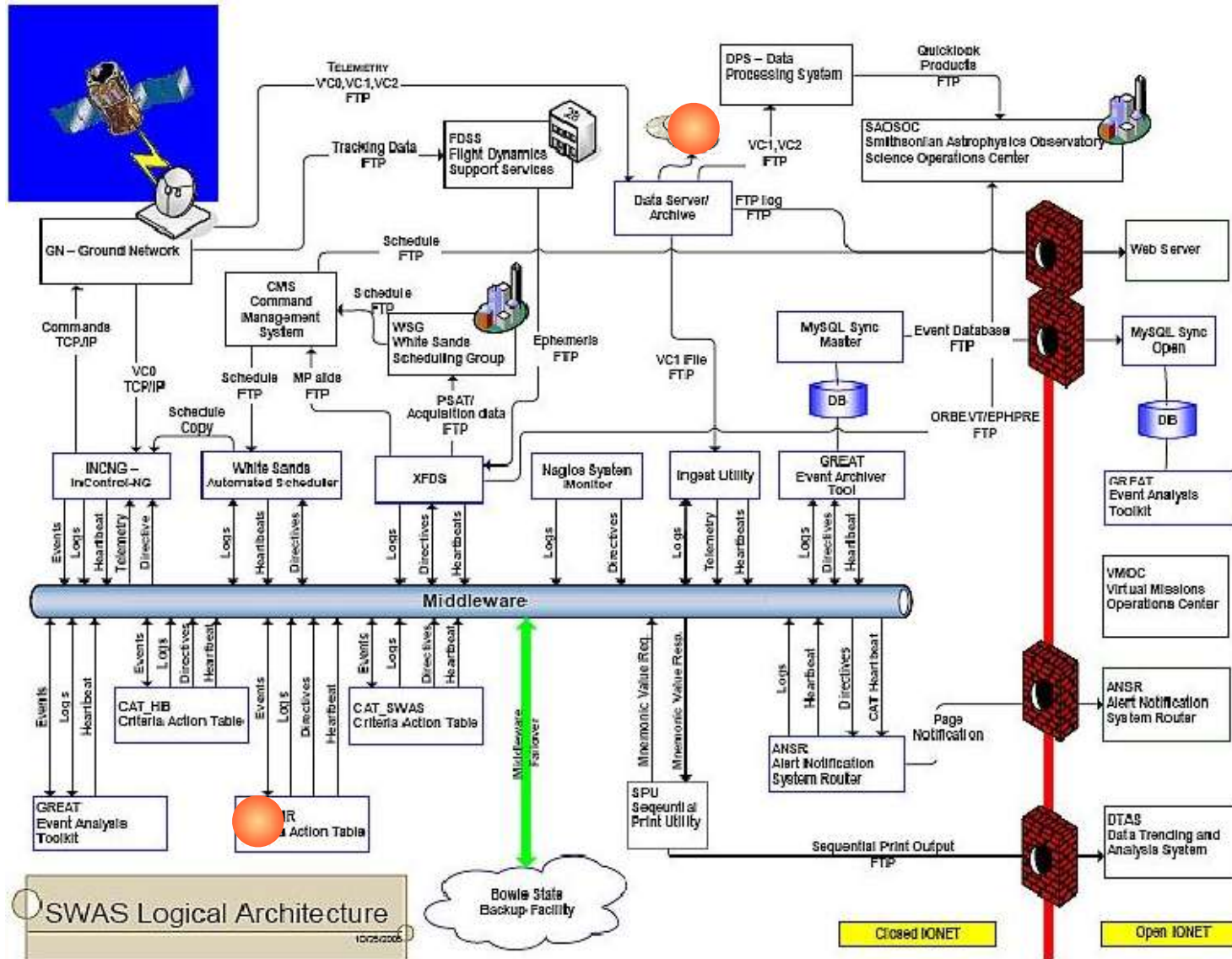
Resiliencia



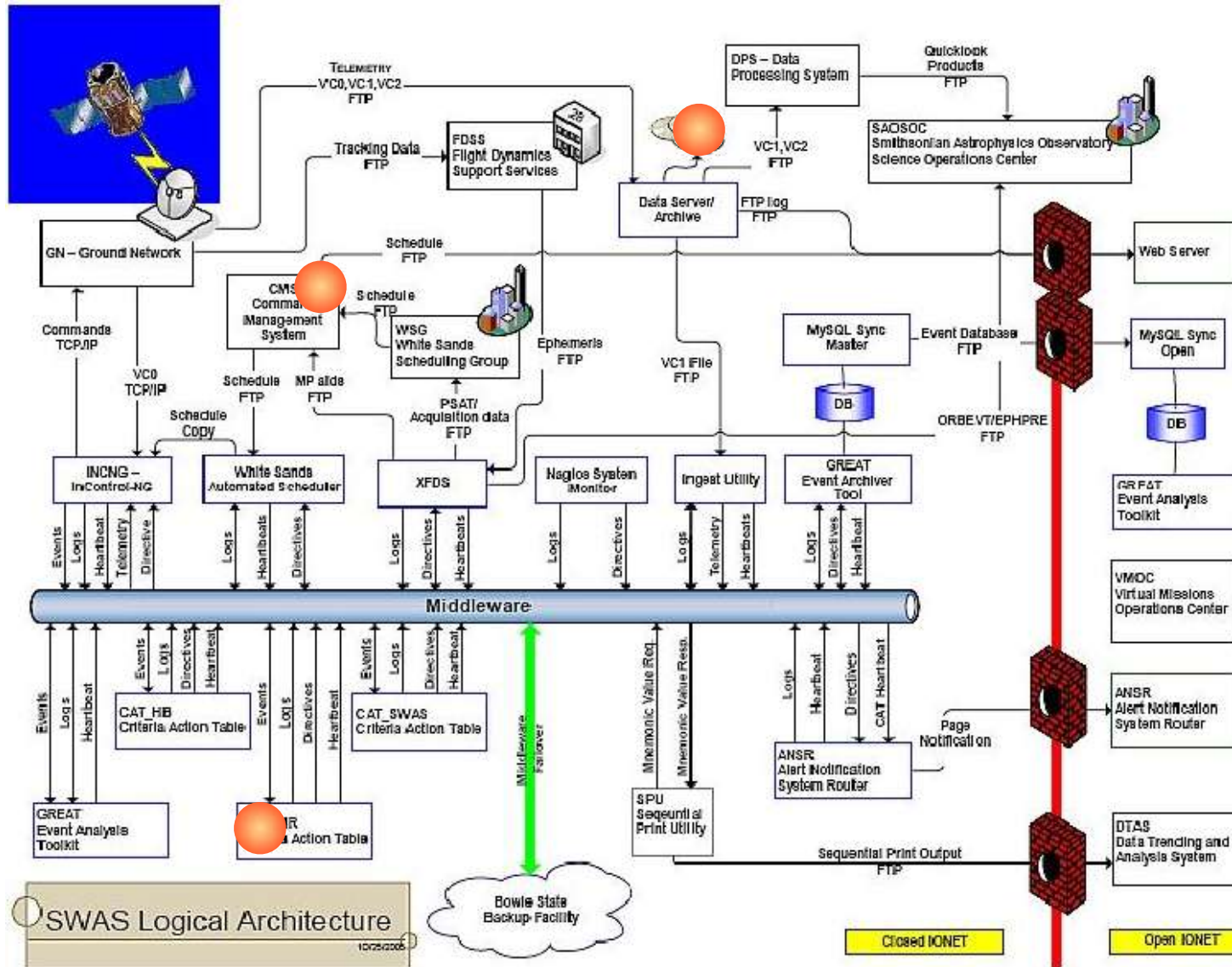
Resiliencia



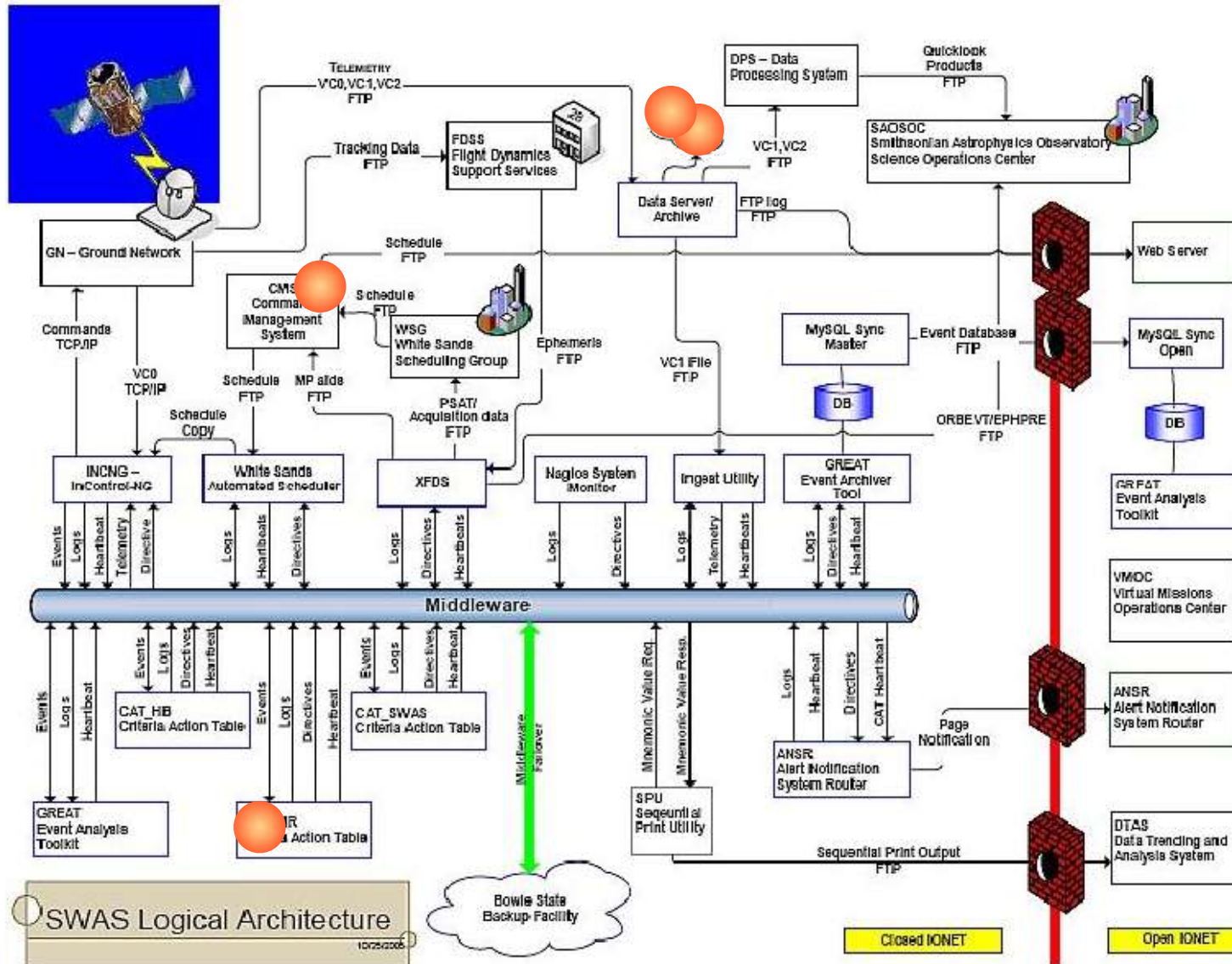
Resiliencia



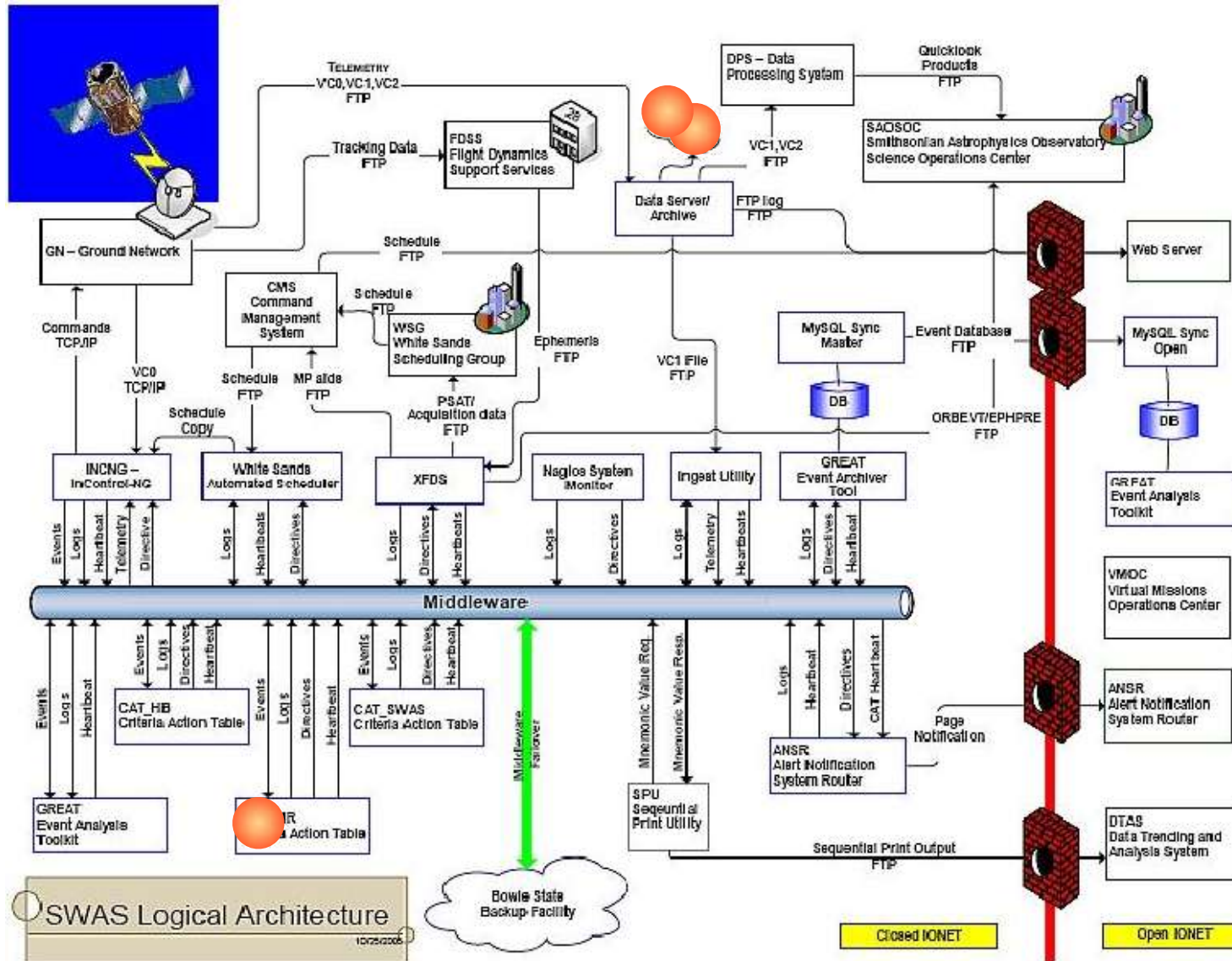
Resiliencia



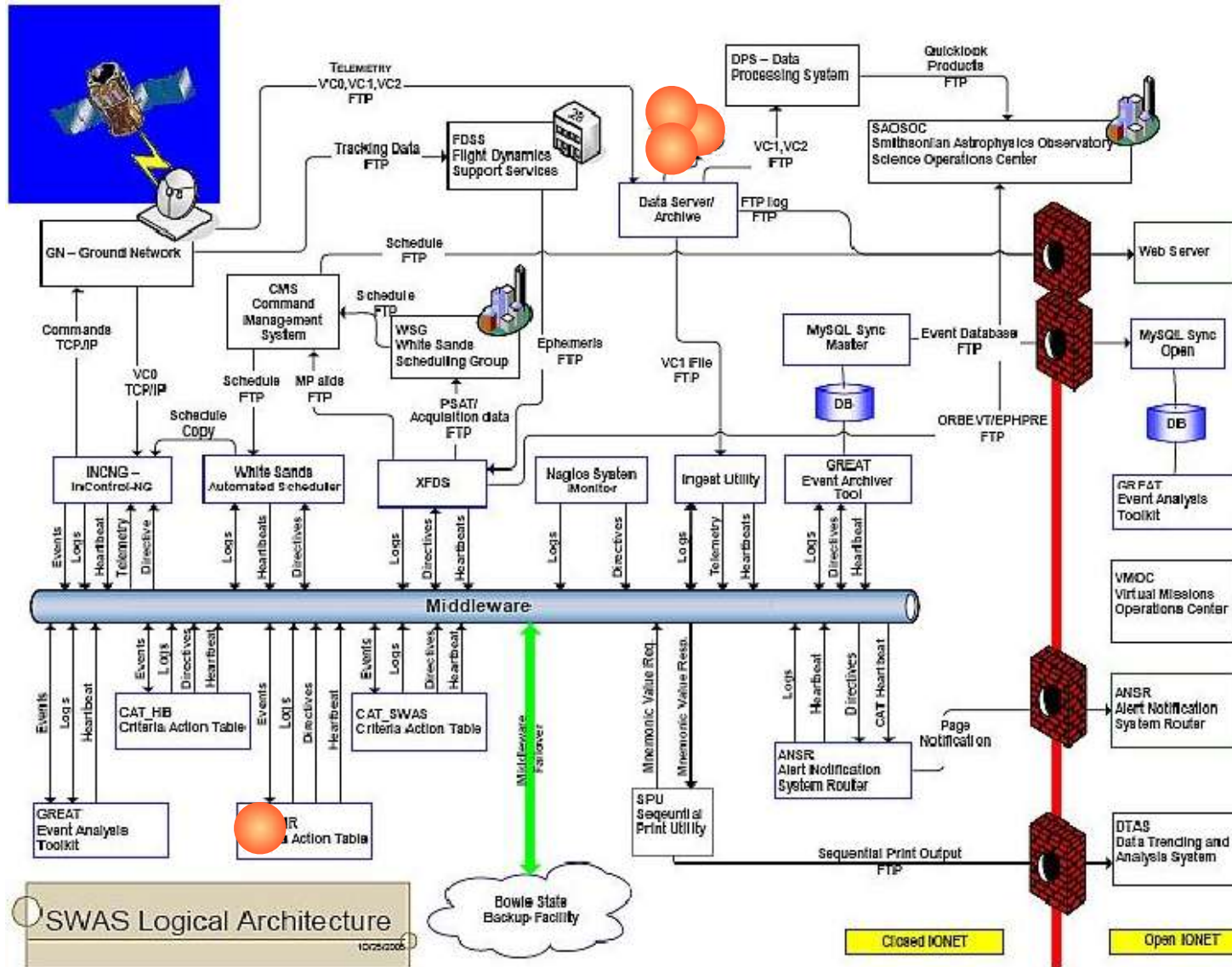
Resiliencia



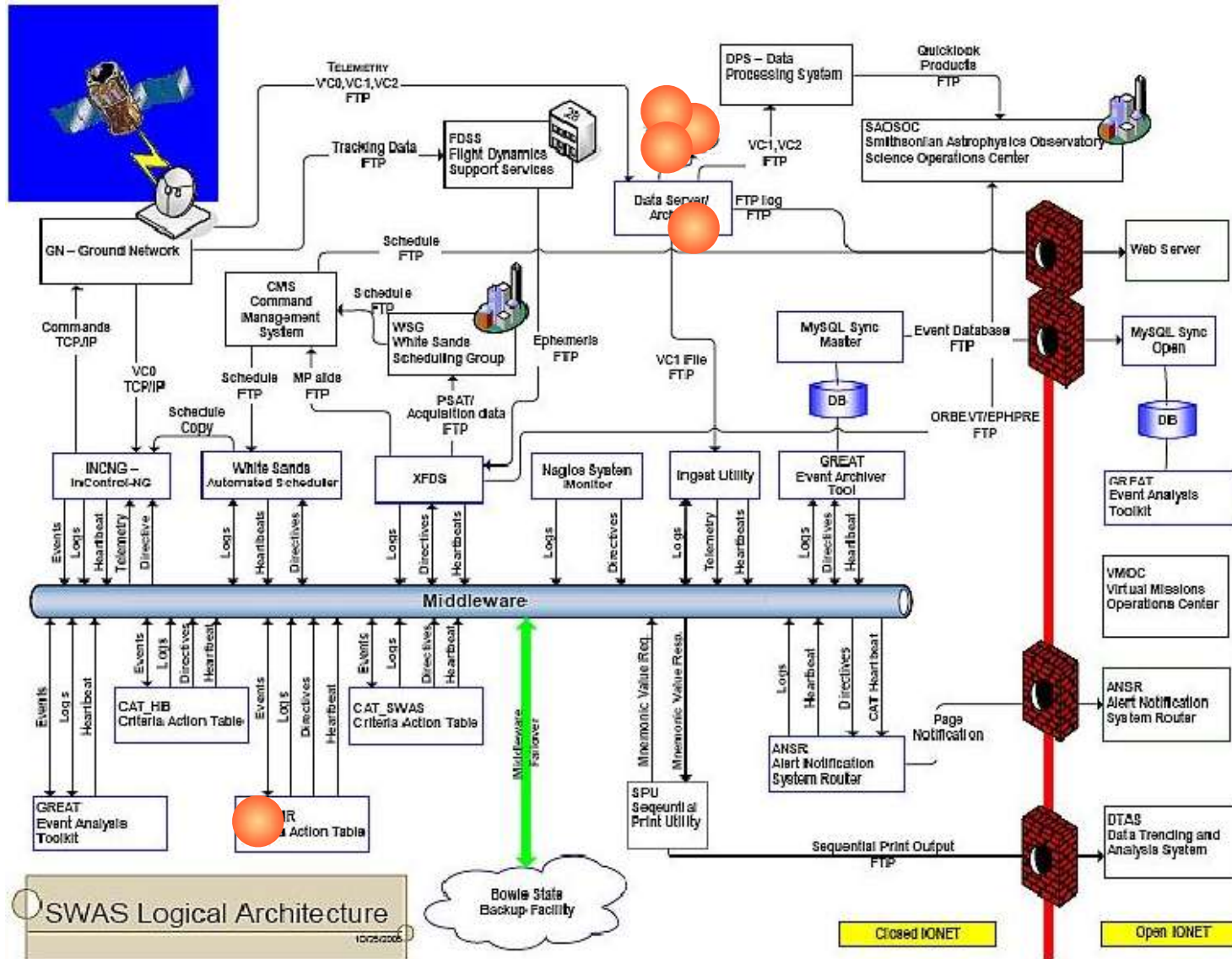
Resiliencia



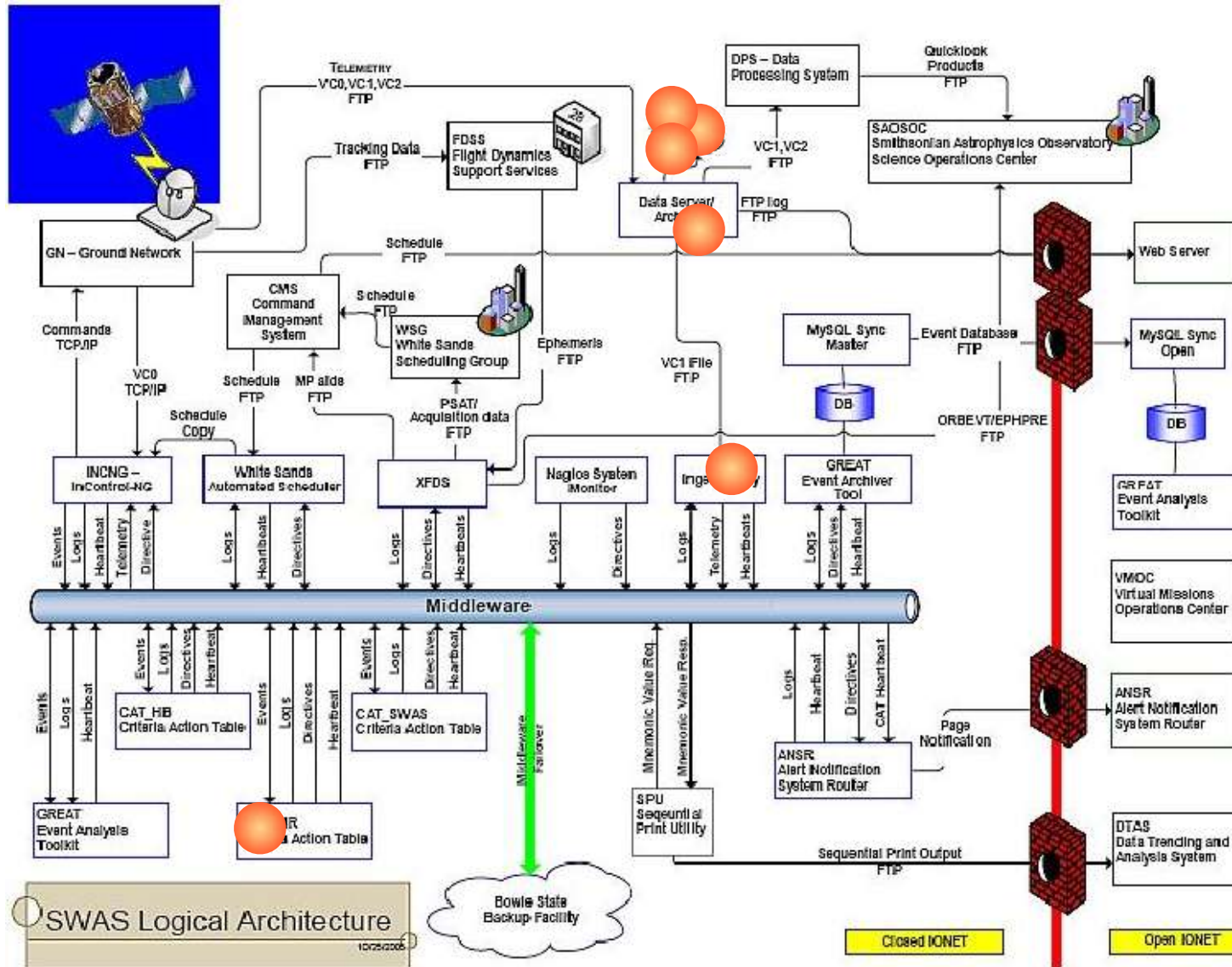
Resiliencia



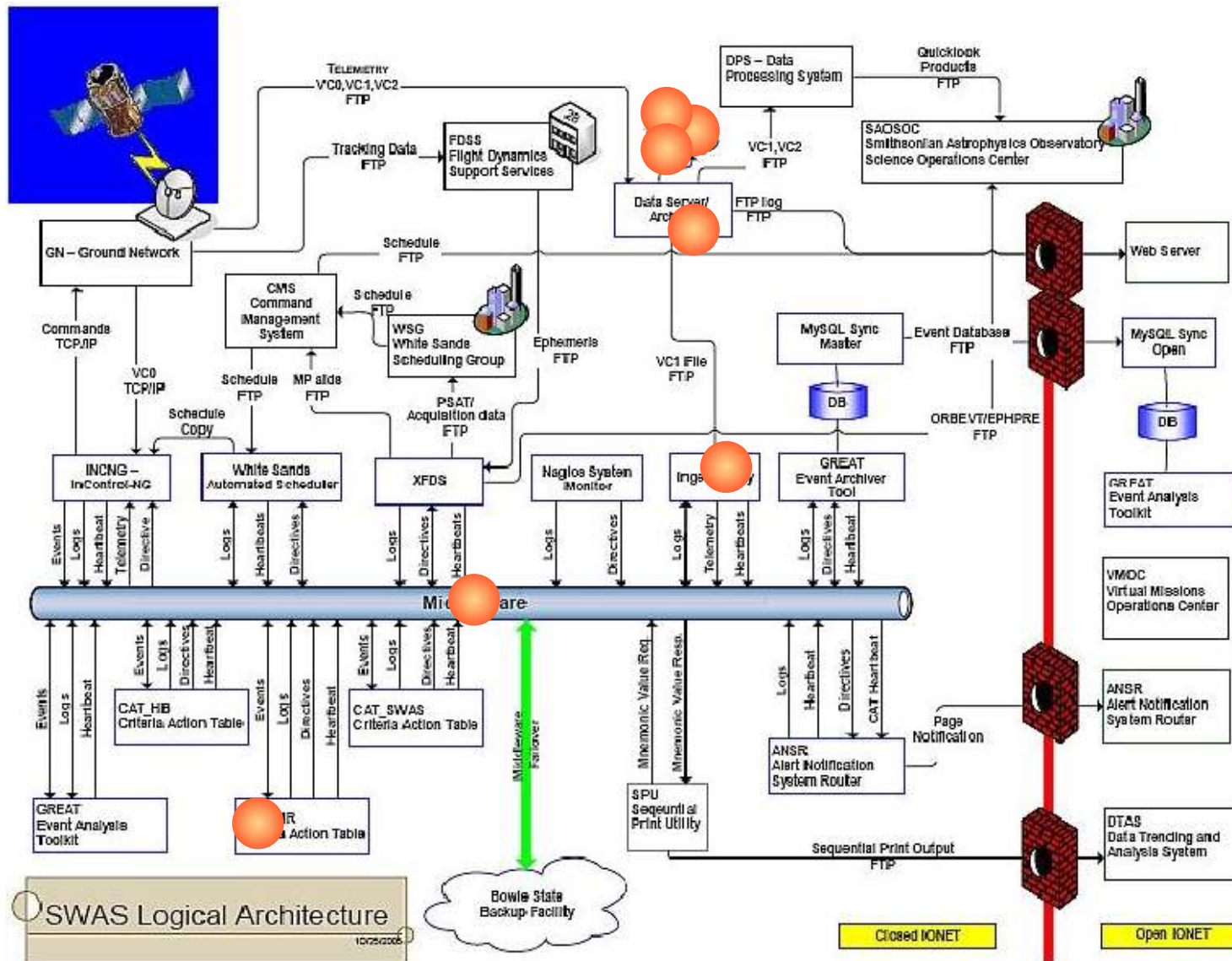
Resiliencia



Resiliencia



Resiliencia



Cómo confrontar las fallas

Redundancia
Redundancia
Redundancia
Redundancia
Redundancia
Redundancia

Cómo confrontar las fallas

1. **Failover:** Varias componentes idénticas de respaldo. Cuando la componente principal falla el sistema lo detecta y cambia a una de las de respaldo.

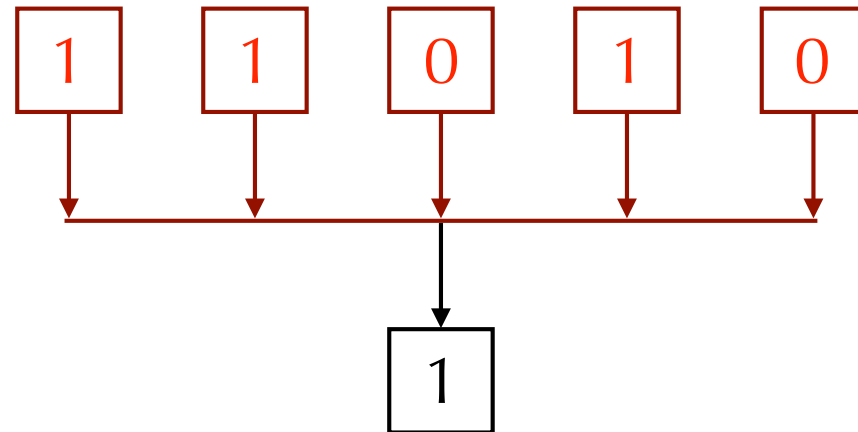


A veces la
reparación es posible

Cómo confrontar las fallas

2. **Votación:** Varias componentes idénticas activas. La información correcta se decide por votación.

Reparación
mediante refrescado
periódico



Cómo confrontar las fallas

3. Detección y corrección de errores:



Detección:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \underbrace{\hspace{10em}} & & & & & & & & \swarrow \\ \underbrace{(0 + 1 + 0 + 1 + 1 + 1 + 0 + 0)}_{\text{mod } 2} = 0 & & & & & & & & \swarrow \end{array}$$



Cómo confrontar las fallas

3. Detección y corrección de errores:



Detección:

0 1 0 1 1 1 0 0 0

$(0 + 1 + 0 + 1 + 1 + 1 + 0 + 0) \bmod 2 = 0$



0 1 0 1 1 0 0 0 0

$(0 + 1 + 0 + 1 + 1 + 0 + 0 + 0) \bmod 2 = 1$



Cómo confrontar las fallas

3. Detección y corrección de errores:



Corrección:

0 1 0 1 1 1 0 0



Cómo confrontar las fallas

3. Detección y corrección de errores:



Corrección:

0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0



Cómo confrontar las fallas

3. Detección y corrección de errores:



Corrección:

0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0

0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 1 0 0



Cómo confrontar las fallas

3. Detección y corrección de errores:



Corrección:

0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0

0 1 0 1 1 1 1 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0



0 1 0 1 1 1 1 0

0 1 0 1 1 0 0

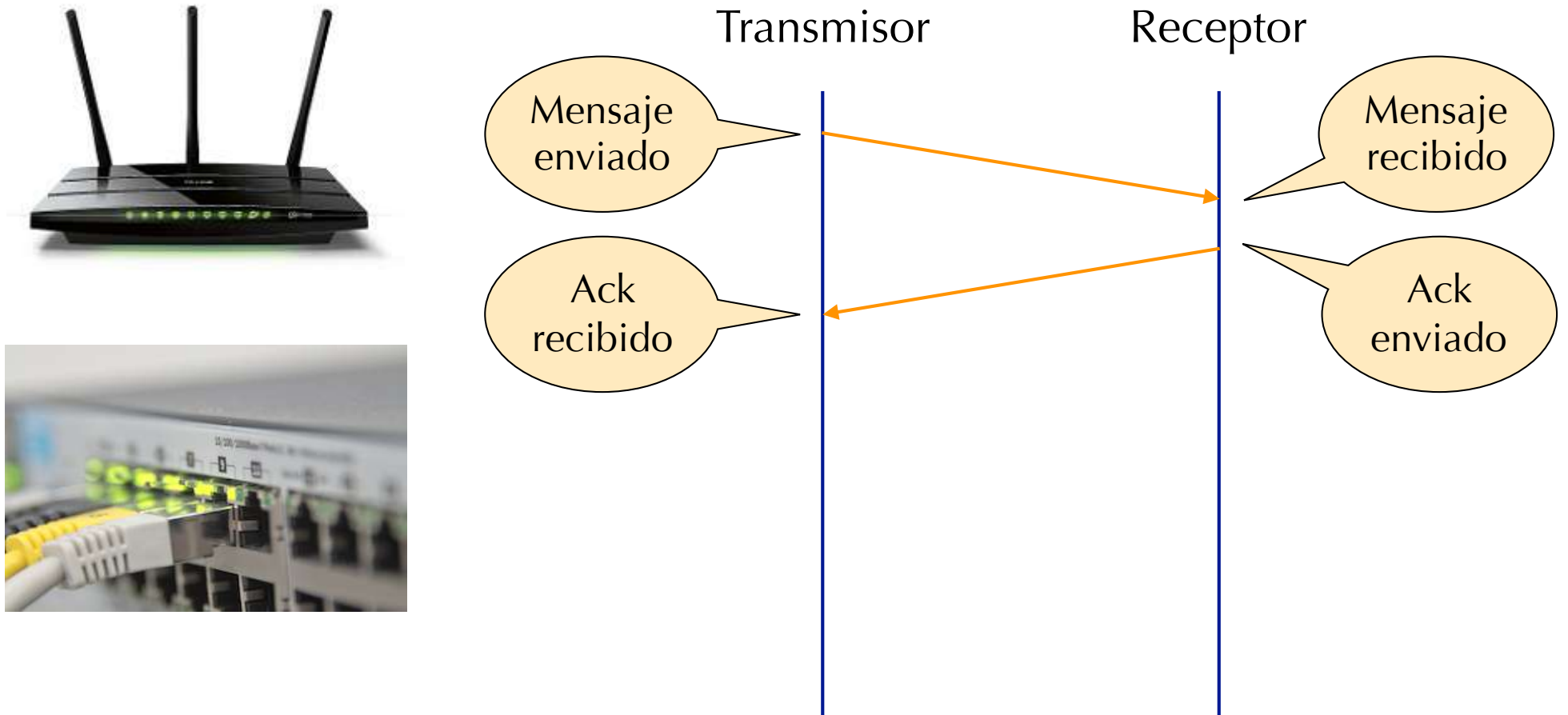
0 1 0 1 1 1 0 0

0 1 0 1 1 1 0 0

Votación
bit a bit

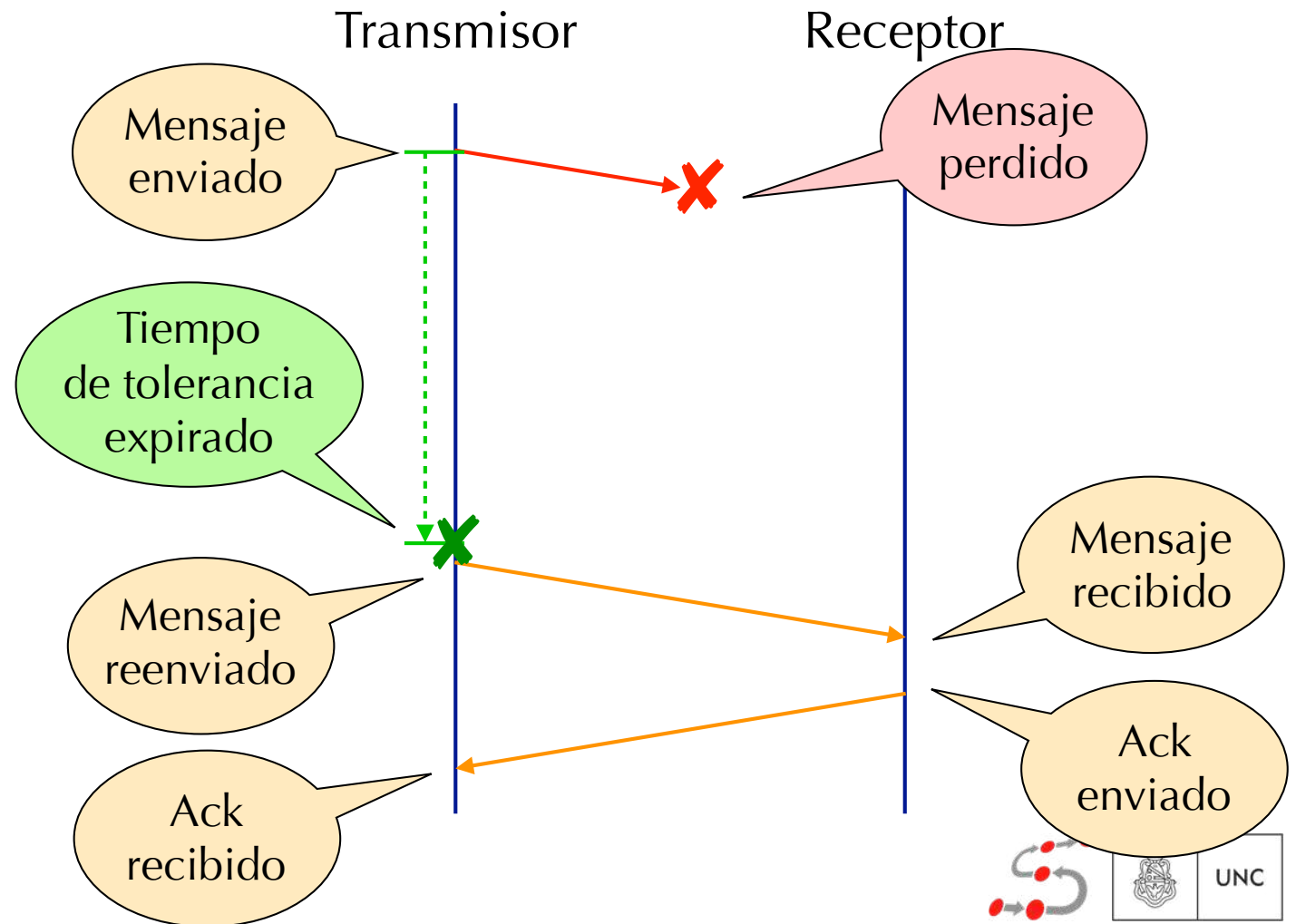
Cómo confrontar las fallas

4. Reconocimientos (Acks) y timeouts:



Cómo confrontar las fallas

4. Reconocimientos (Acks) y timeouts:



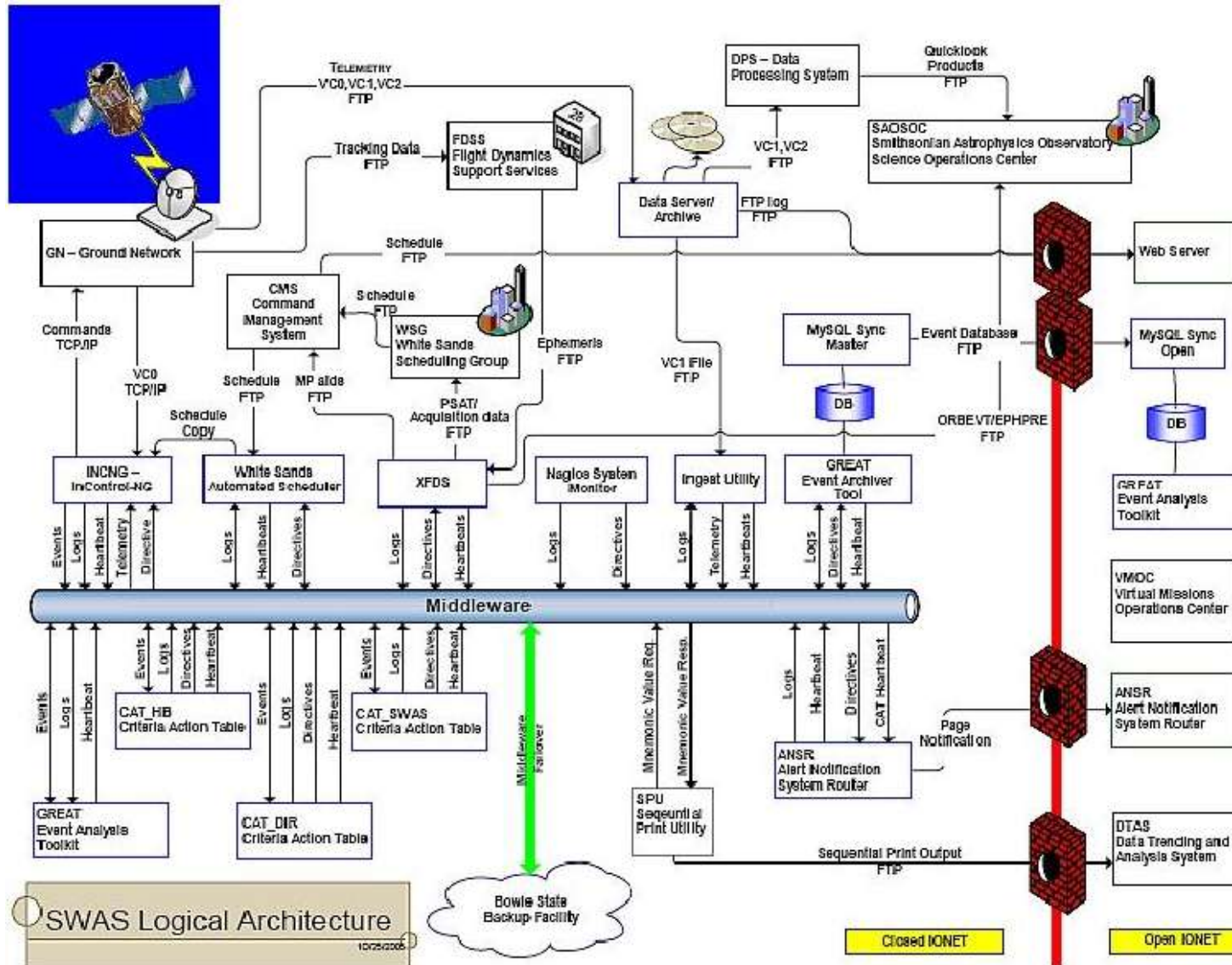
Eventos

Los eventos pueden cuantificarse probabilísticamente

Ejemplos:

- ❖ Probabilidad de pérdida de un mensaje
- ❖ Tiempo esperado de vida de una fuente de alimentación
- ❖ Tiempo esperado de reparación del disco rígido
- ❖ Tiempo esperado de transmisión tierra-satélite
- ❖ Probabilidad de alteración de un bit bajo radiación
- ❖ Tiempo esperado de refrescado de memoria

Resiliencia



Análisis de resiliencia

- ❖ Se requiere **gran confiabilidad**.

Ej: ❖ Disponibilidad del 99.999% (“cinco nueves”)

- ❖ Tiempo esperado de falla de 6 años

- ❖ Determinadas por **eventos raros**:

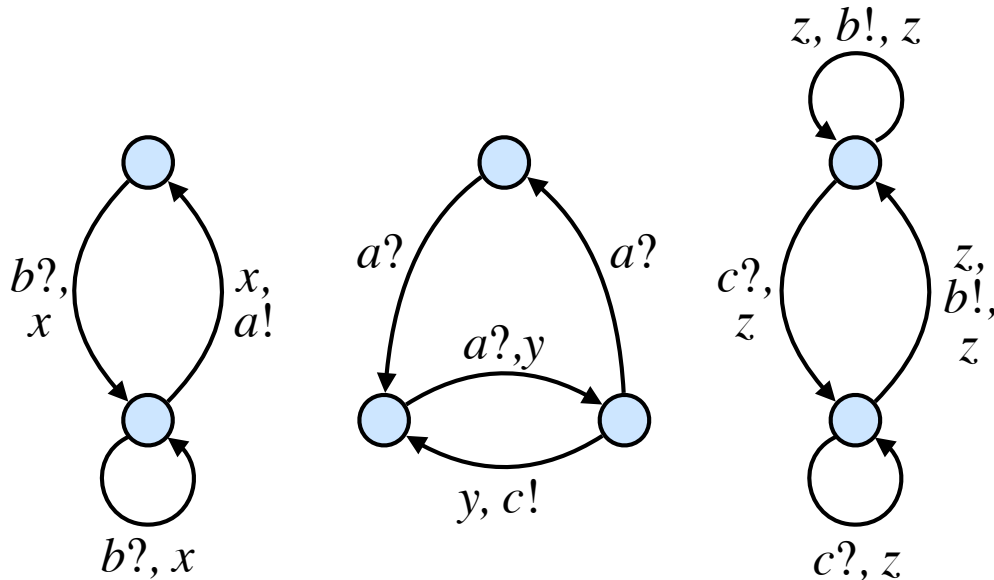
El evento combinado que determina esa tipo de propiedades suele tener probabilidad muy baja de ocurrir.

Análisis de resiliencia

Un par de desafíos

Modelado composicional

Simulación de eventos raros



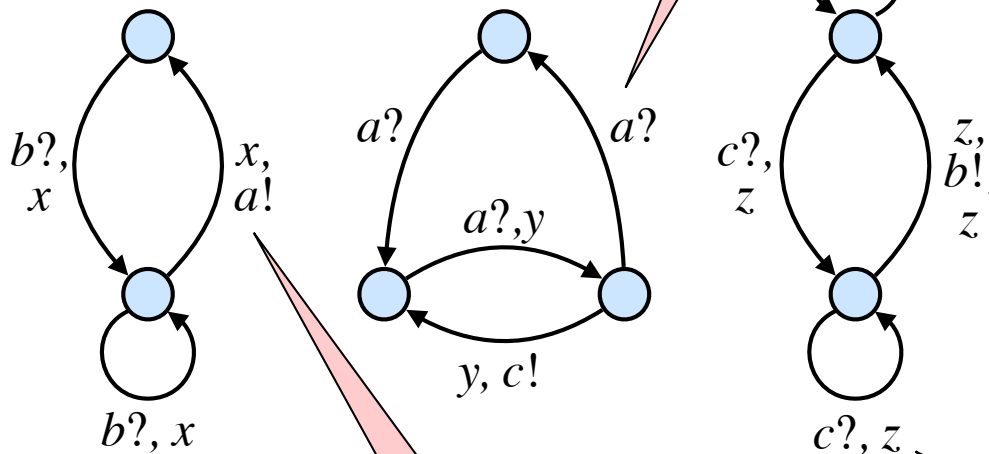
Redes de
autómatas estocásticos
con entradas y salidas
(IOSA)

Análisis de resiliencia

Un par de desafíos

Modelado compo

Simulación de eventos raros



Evento de entrada

Evento de salida

auton. estocásticos
con entradas y salidas
(IOSA)

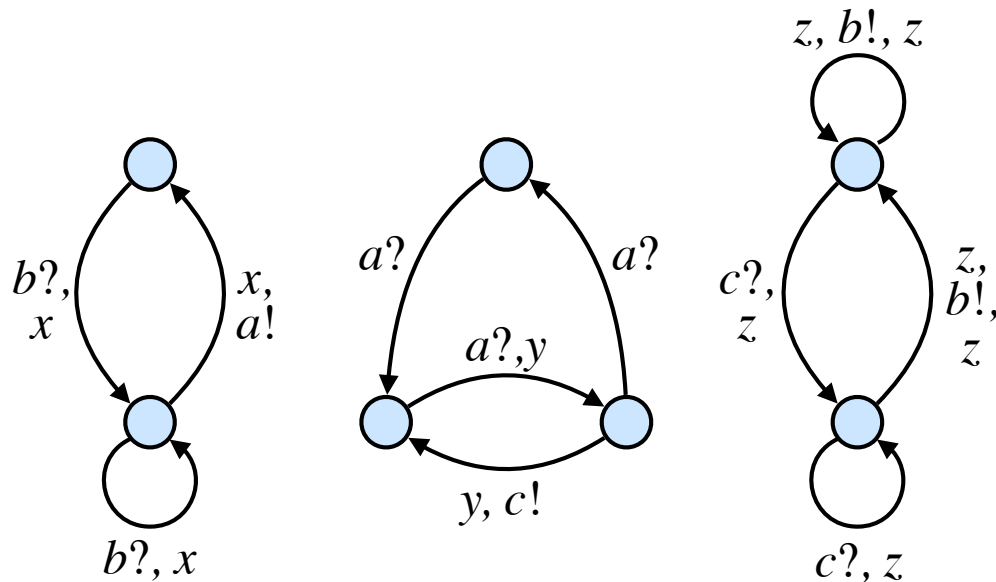
La salida $b!$ se ejecuta cuando el timer z se decrementa a 0

Asignación de un valor aleatorio al timer z

Análisis de resiliencia

Un par de desafíos

Modelado composicional



Redes de
autómatas estocásticos
con entradas y salidas
(IOSA)

Simulación de eventos raros

- ❖ Preparado para **Simulación por Eventos Discretos**
- ❖ El **problema** de realizarla siguiendo la técnica de simulación Monte Carlo

...

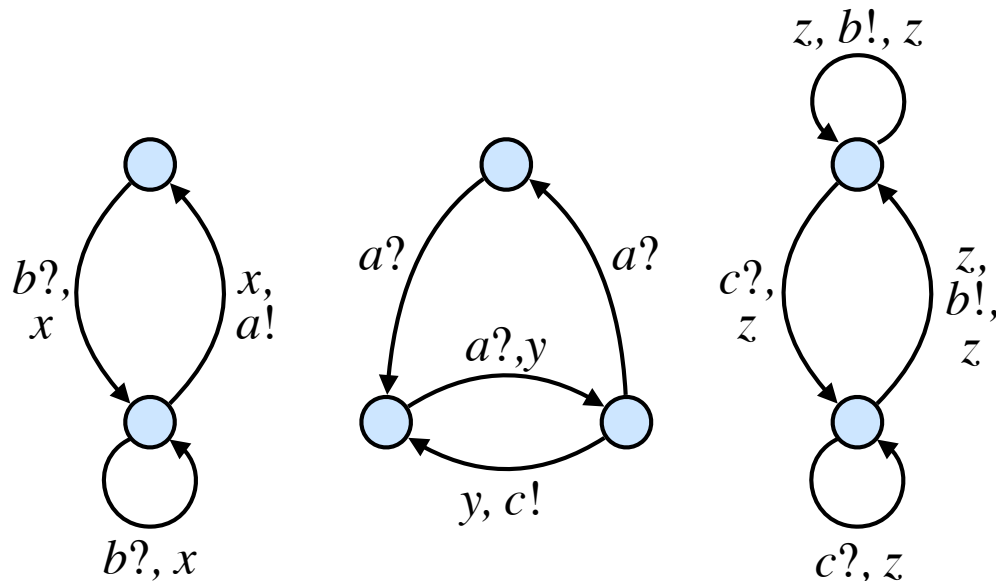
Análisis de resiliencia

Raúl E. Monti

Un par de desafíos

Carlos E. Budde

Modelado composicional



Redes de
autómatas estocásticos
con entradas y salidas
(IOSA)

Simulación de eventos raros

- ❖ Preparado para **Simulación por Eventos Discretos**
- ❖ El **problema** de realizarla siguiendo la técnica de simulación Monte Carlo

...

Holger Hermanns
Matías D. Lee
Damián Barsotti

Resiliencia en Sistemas Computacionales Complejos

Pedro R. D'Argenio

Dependable Systems Group
FaMAF, UNC - CONICET

<http://dsg.famaf.unc.edu.ar>



Conferencia Gaviola, Noviembre 2015

