Task Force on TARGET Instant Payments Settlement

5th Task Force Meeting
9 March 2017

Deployment management approach for infrastructure/software changes to comply with no downtime
TIPS availability

- TIPS will operate 24/7/365, without the need for any maintenance window or daily interruption of the service.

- The following slides explain how TIPS should manage any potential infrastructure or software changes (including ISO release) without stopping the service. The process applies:
  - Into TIPS server
  - Within TIPS community
    - E.g.: change of ISO 20022 standard.
Deployment of updates in a 24/7 environment with no planned downtime

Change management process – Step 1

- Infrastructure changes: Maintenance, Patch management, etc.
- Software changes: Deployment of new versions ISO standard, Static data injection, etc.

Backward compatibility or Stability: it is a precondition
Deployment of updates in a 24/7 environment with no planned downtime

**Change management process – Step 2**

“Not eligible messages”: Messages are not used for output (used for comparison?)

E.g.: New configured instance must be really active only after some checks of correctness have been done.
Deployment of updates in a 24/7 environment with no planned downtime

**Change management process – Step 3**

“New version” messages: In this phase both OLD and NEW version messages can be delivered to PSPs.
Deployment of updates in a 24/7 environment with no planned downtime

Change management process – Step 4

“New version” messages:
In case of error or unsuccessful deployment, the node is rolled-back to the original status.
To allow continuous operation:

- Any change in the system should be designed to be deployed in two steps:
  
  - **Step 1 – Compatibility mode** where new and old behaviors can coexist.
  
  - **Step 2 – New function mode** where old behavior is not allowed anymore.
Deployment of updates in a 24/7 environment with no planned downtime

**Compatibility mode 2/3**

Example: A new mandatory tag must be added to a message from time $T_F$ change in the system should be designed to be deployed in two steps

- **Step 1 – Compatibility mode:**
  - At $T_0$ all receivers must be able to accept (maybe ignoring it) the new tag
  - At $T_1 > T_0$ senders can produce the new tag

- **Step 2 – New function mode:**
  - At $T_F > T_1$ all senders must produce the new tag
  - Old format not allowed anymore
Compatibility mode 3/3

Example: A new service must be provided starting from time $T_F$

- **Step 1 – Compatibility mode:**
  - At $T_0$ all servers must be able to provide the service if asked to do so. No clients can require the new service

- **Step 2 – New function mode:**
  - At $T_F > T_0$ any client can require the new service