TTTechAuto

MotionWise TaskMonitoring & Watchdog

Partner Enablement Training

CPI-I SSP / Srikrishna Balaji Prayaga

October 30, 2023

Agenda

TTTechAuto

TTTechAuto

01

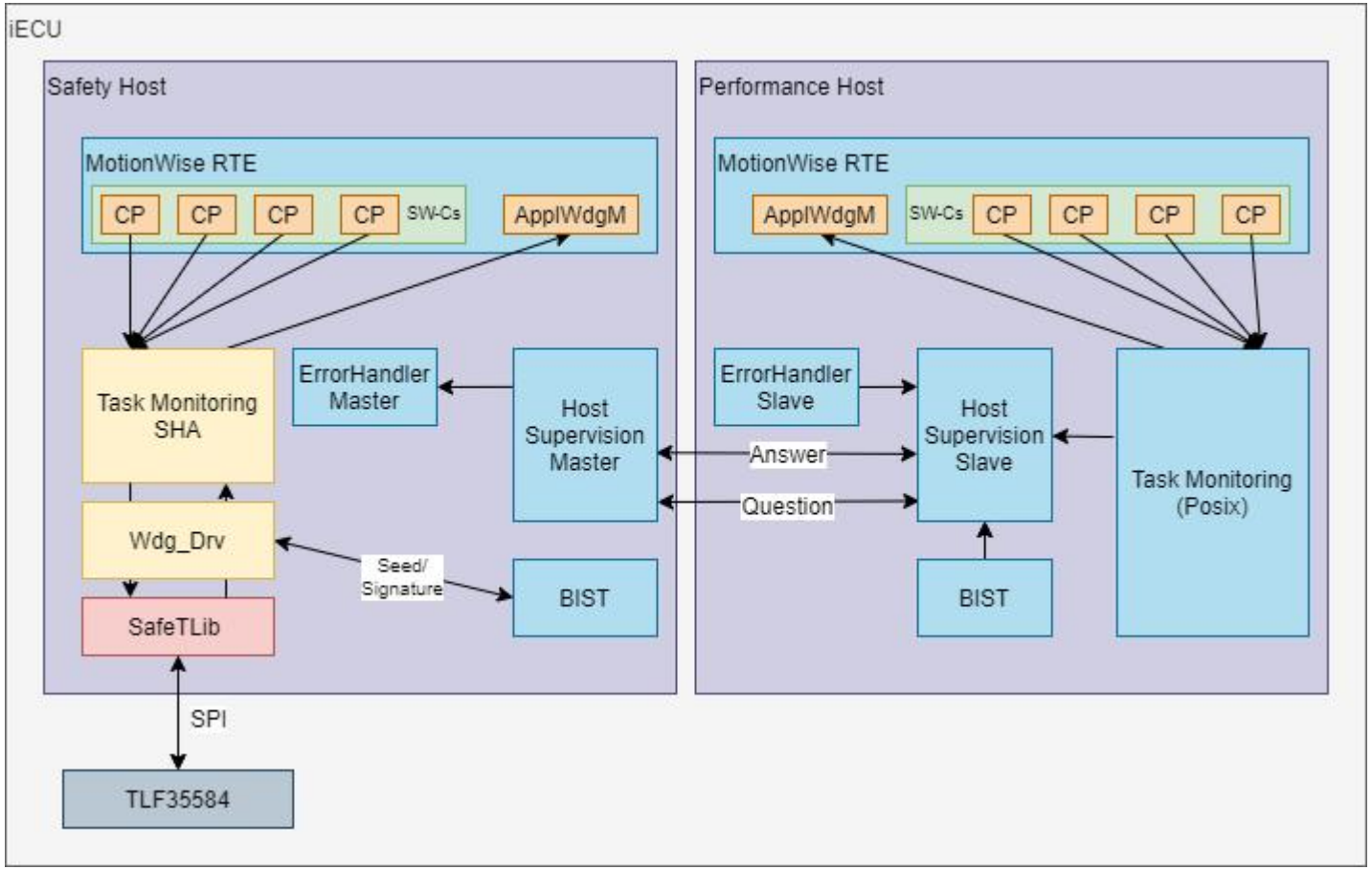
Overview

Task Monitoring and Watchdog

October 30, 2023

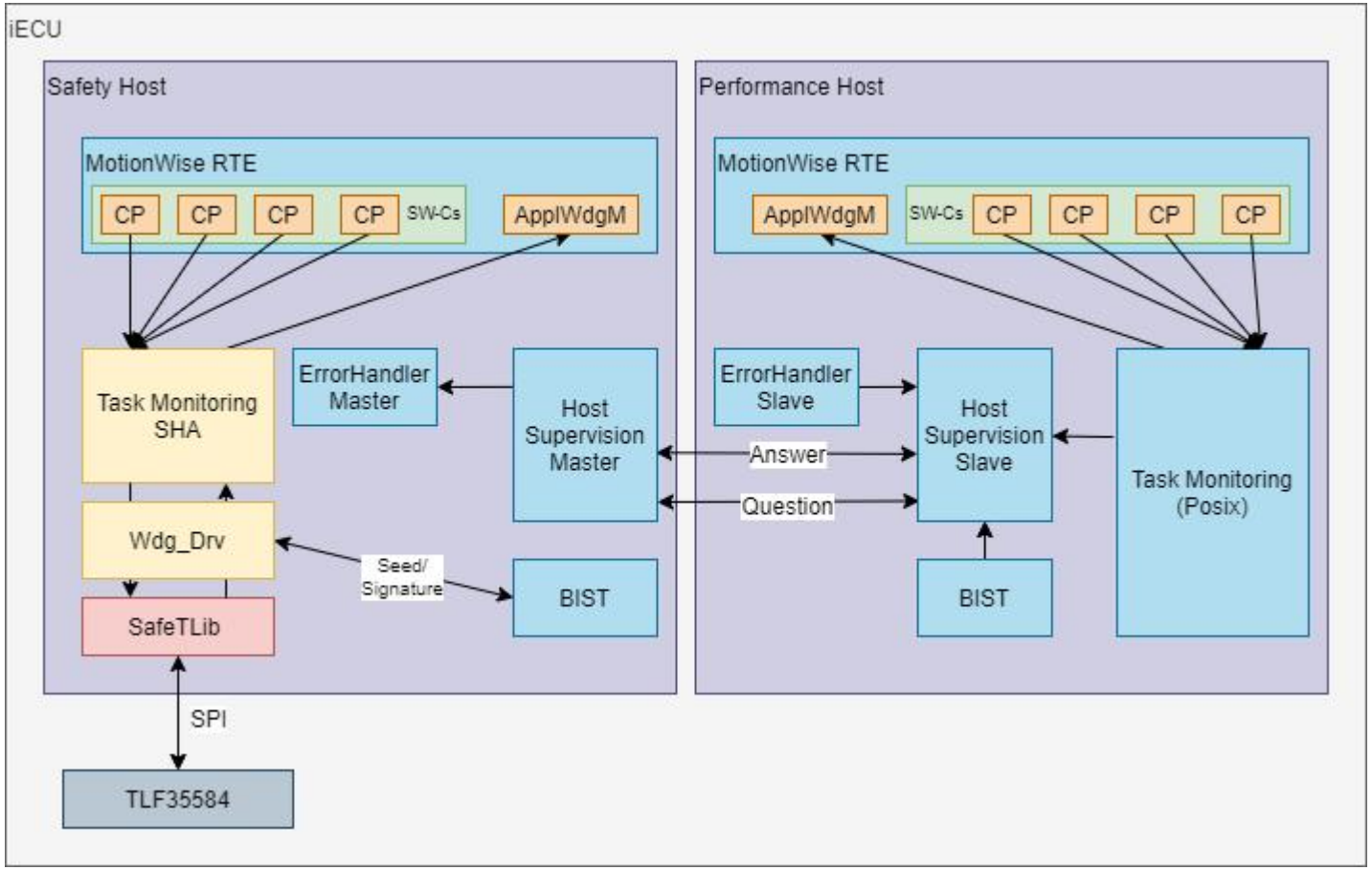
Task Monitoring and Watchdog - Context

1. Every host in a MotionWise ECU is monitored by an external entity.
2. The Performance Host is monitored by the Safety Host.
3. The Safety Host is monitored by an external Watchdog TLF35584.
4. The monitoring has two aspects:
    1. Temporal monitoring of the software - *Task Monitoring Service*
    2. Monitoring of HW health – *BIST Service*
5. The result of the monitoring is then communicated to the external monitoring entity.

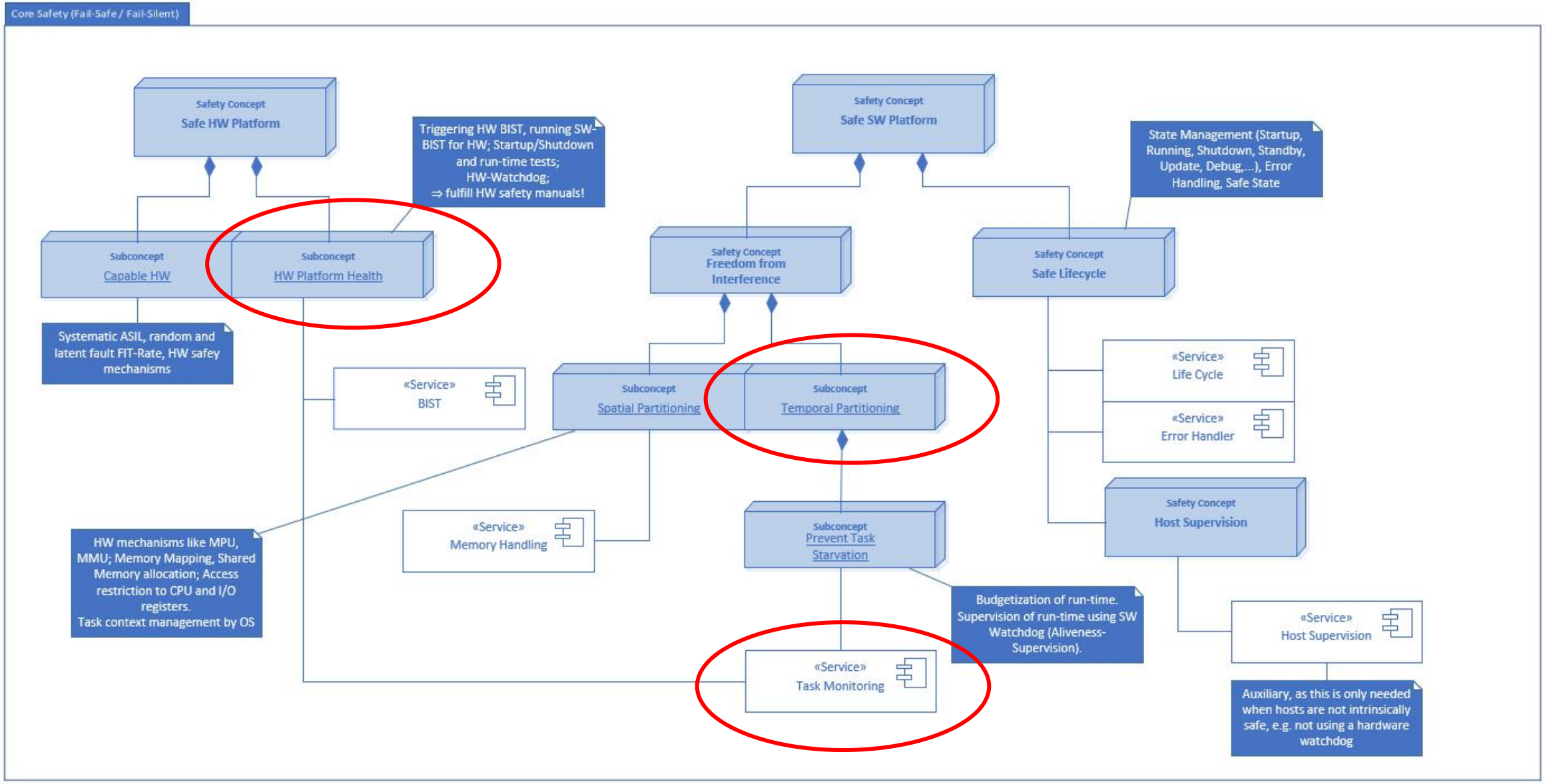


Task Monitoring and Watchdog - Scope

- The scope of this training session is:
  1. Temporal Monitoring on the Performance Host – *Task Monitoring Posix*
  2. Temporal Monitoring on the Safety Host – *Task Monitoring SHA*
  3. Communication between Safety Host and the external watchdog TLF35584 – *Wdg_Drv* and *SafeTLib*



Task Monitoring and Watchdog – in the big picture…



TTTechAuto

02

Functionality

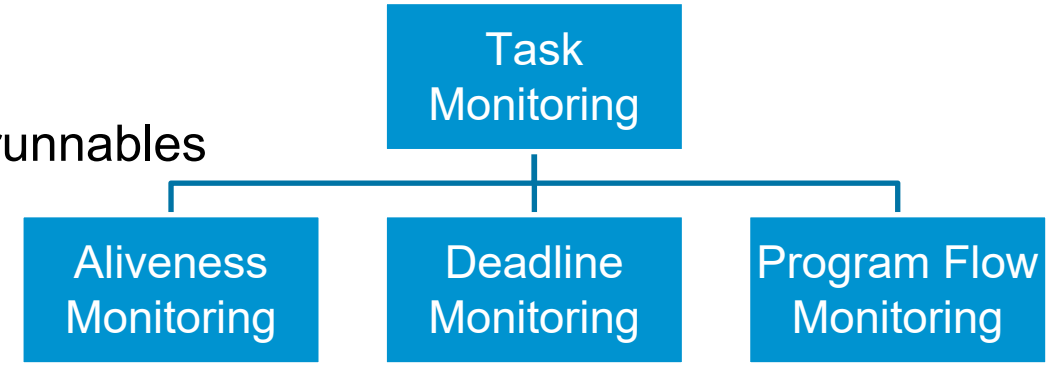
Task Monitoring and Watchdog

October 30, 2023

Task Monitoring - Functionality

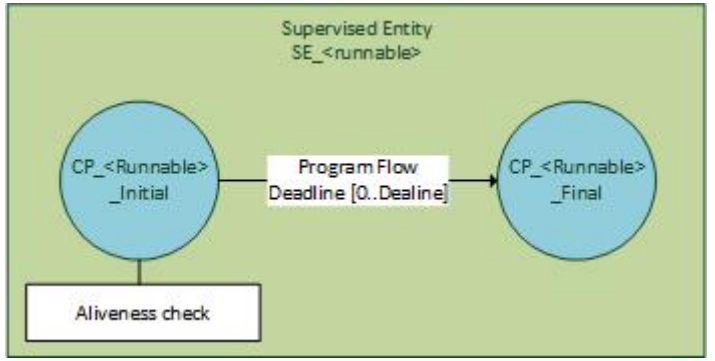
TTTechAuto

1. Task Monitoring Service achieves the following on Safety Host and Performance Host:
    1. Detect violations by monitoring-
        - execution time violations of scheduled tasks/runnables
        - task starvation
    2. Trigger safety reaction-
        - if violations occur



Supervised Entities

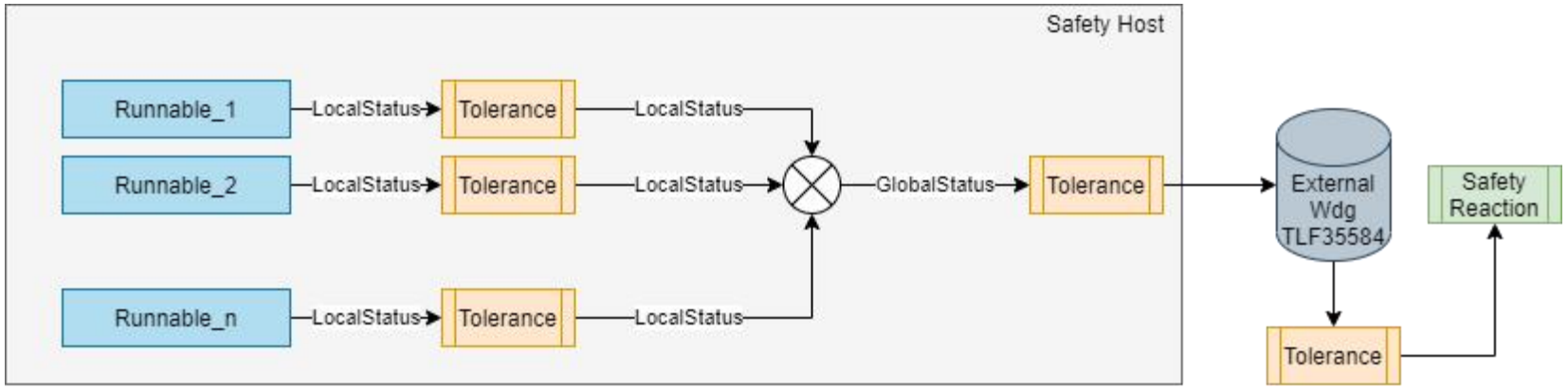
1. Runnables and tasks are Supervised Entities.
2. Each SE logs its initial and final checkpoints, before and after runnable execution, respectively.
3. Monitoring is performed on the checkpoints.
4. Aliveness monitoring is performed on the initial checkpoints.
5. Deadline and ProgramFlow monitoring make use of both initial and final checkpoints.



Note: 'Task Monitoring' is different from 'Time Monitoring', which is achieved using the service TimeSyncMonitoring.

TTTechAuto

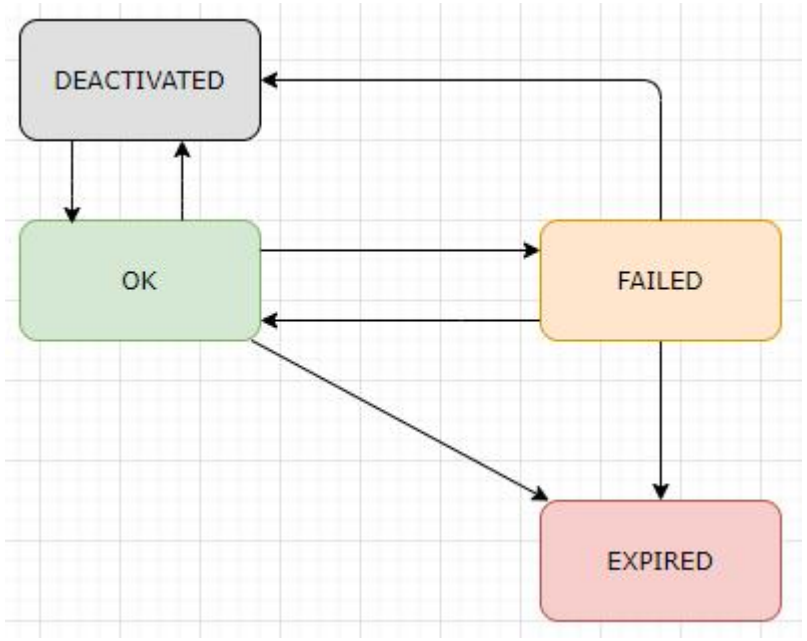
Stages of monitoring and configurable tolerances

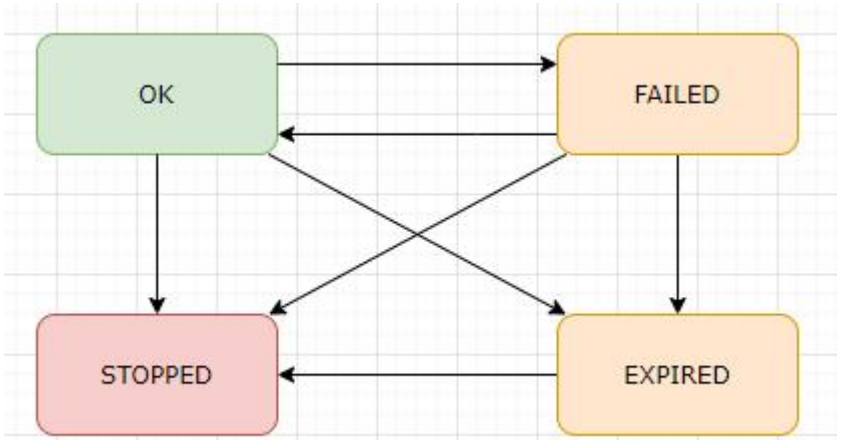


1. Each monitored runnable (also known as Supervised Entity) has a Local Status.
2. The Local Status of all runnables is combined to form the Global Status.
3. GlobalStatus is an input for triggering safety reaction from the external watchdog.
4. Tolerances can be configured at three different stages:
   - Local Status evaluation
   - Global Status evaluation
   - TLF35584 monitoring

TTTechAuto

Local Status and Global Status





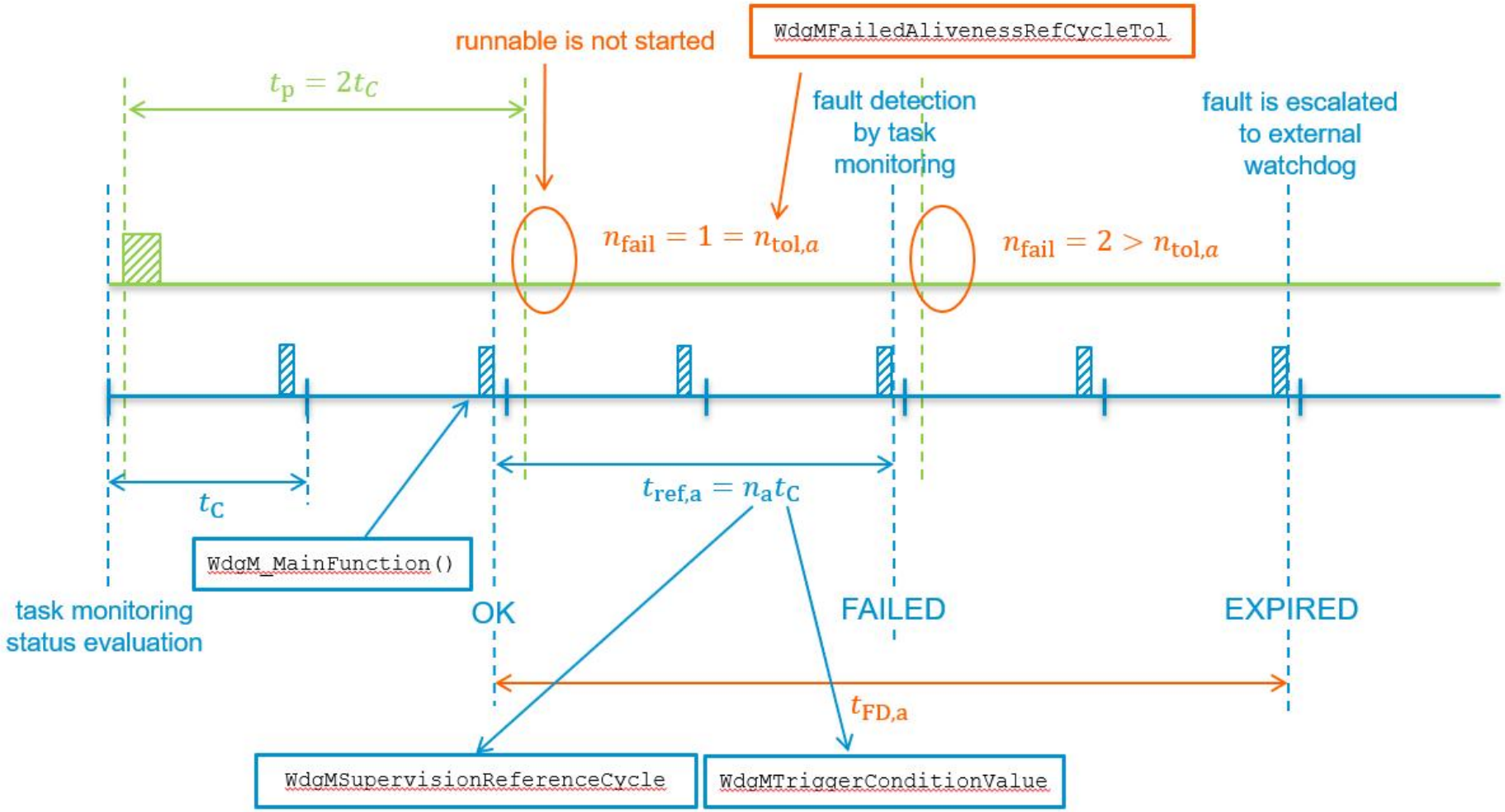
Global Status

- LocalStatus and GlobalStatus start with the status OK.
- GlobalStatus is FAILED, if atleast one LocalStatus is FAILED.
- GlobalStatus is EXPIRED, if atleast one LocalStatus is EXPIRED.
- GlobalStatus is STOPPED, when tolerance on GlobalStatus is exceeded.

TTTechAuto

Aliveness Monitoring

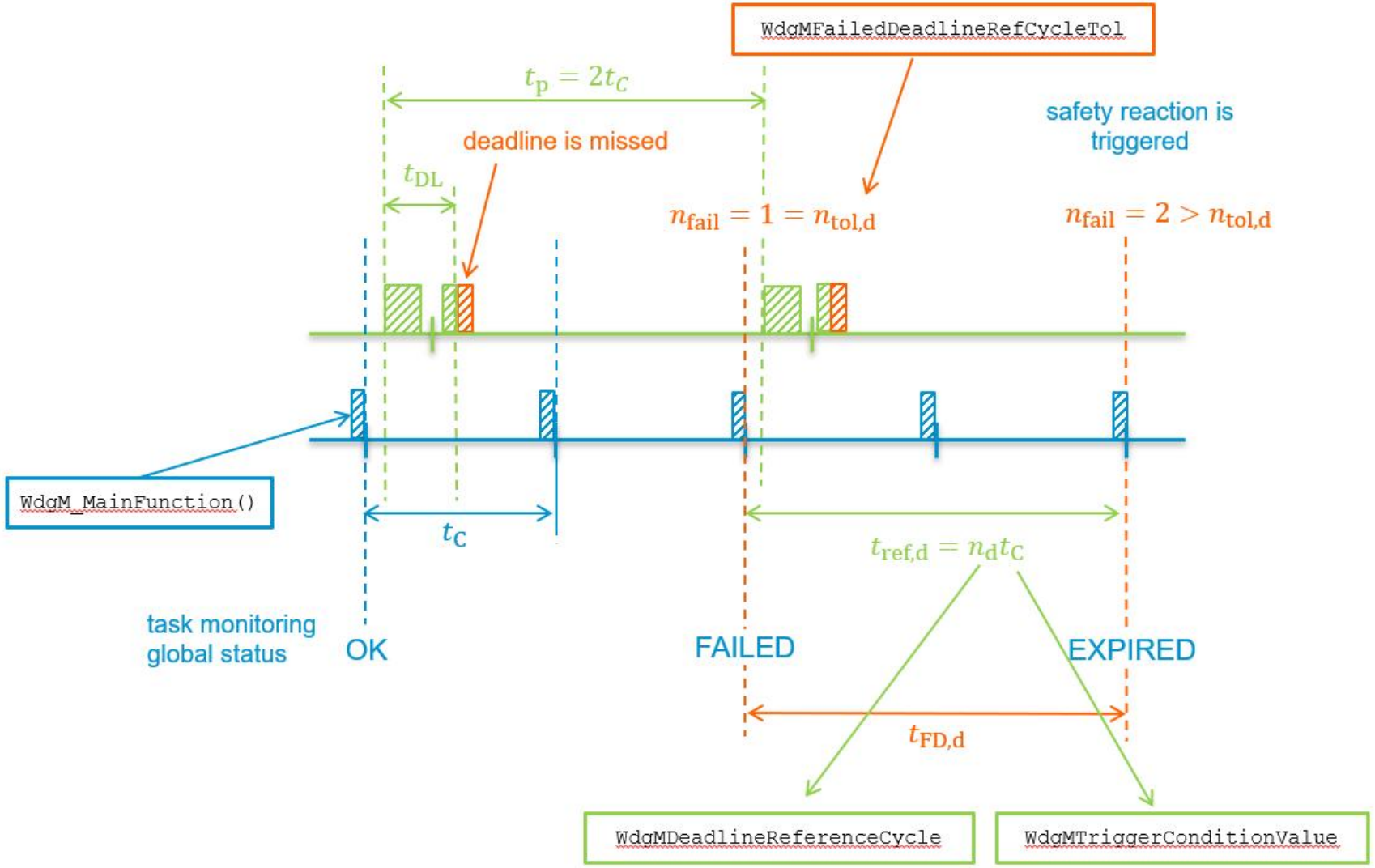
$$t_{FD,a} < (n_{tol,a} + 1)\, t_{ref,a} = (\text{TmFailedAlivenessRefCycleTol} + 1)\text{TmAlivenessReferenceCycle} \times \text{TmTriggerWindowEnd}$$
$$= (\text{WdgMFailedSupervisionRefCycleTol} + 1)\text{WdgMSupervisionReferenceCycle} \times \text{WdgMTriggerConditionValue}$$



TTTechAuto

Deadline Monitoring

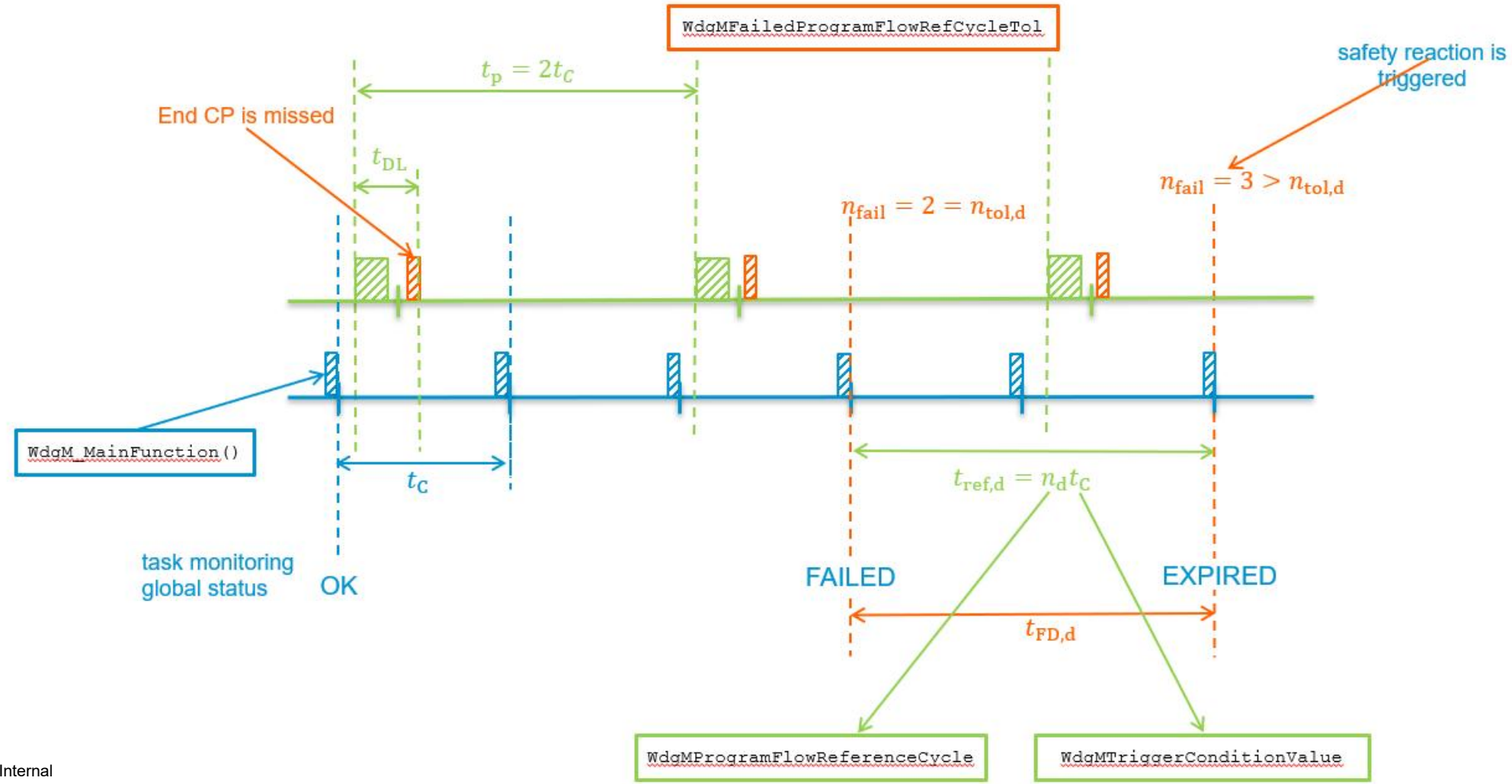
$$t_{FD,d} < (n_{tol,d} + 1)\, t_{ref,d} = (\text{TmFailedDeadlineRefCycleTol} + 1)\text{TmDeadlineReferenceCycle} \times \text{TmTriggerWindowEnd}$$
$$= (\text{WdgMFailedDeadlineRefCycleTol} + 1)\text{WdgMDeadlineReferenceCycle} \times \text{WdgMTriggerConditionValue}$$



TTTechAuto

Program Flow Monitoring

$$tFD,p < (WdgMFailedProgramFlowRefCycleTol + 1)*(WdgMProgramFlowReferenceCycle+1)*WdgMTriggerConditionValue$$

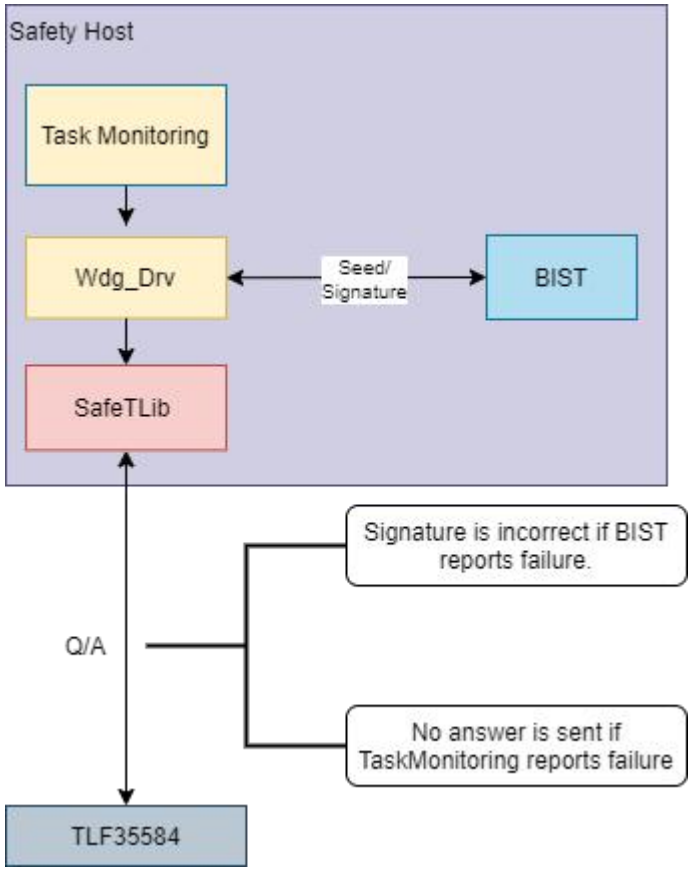


External Watchdog TLF35584

1. TLF35584 is configured to act as an external Functional Watchdog (FWD) to the Safety Host.
2. It expects a servicing from the Safety Host every 10ms.
3. Configured tolerance: 15ms for every servicing, three consecutive failures.
4. On failure detection, TLF35584 activates the SS1 and SS2 lines, and resets the Safety Host Aurix.

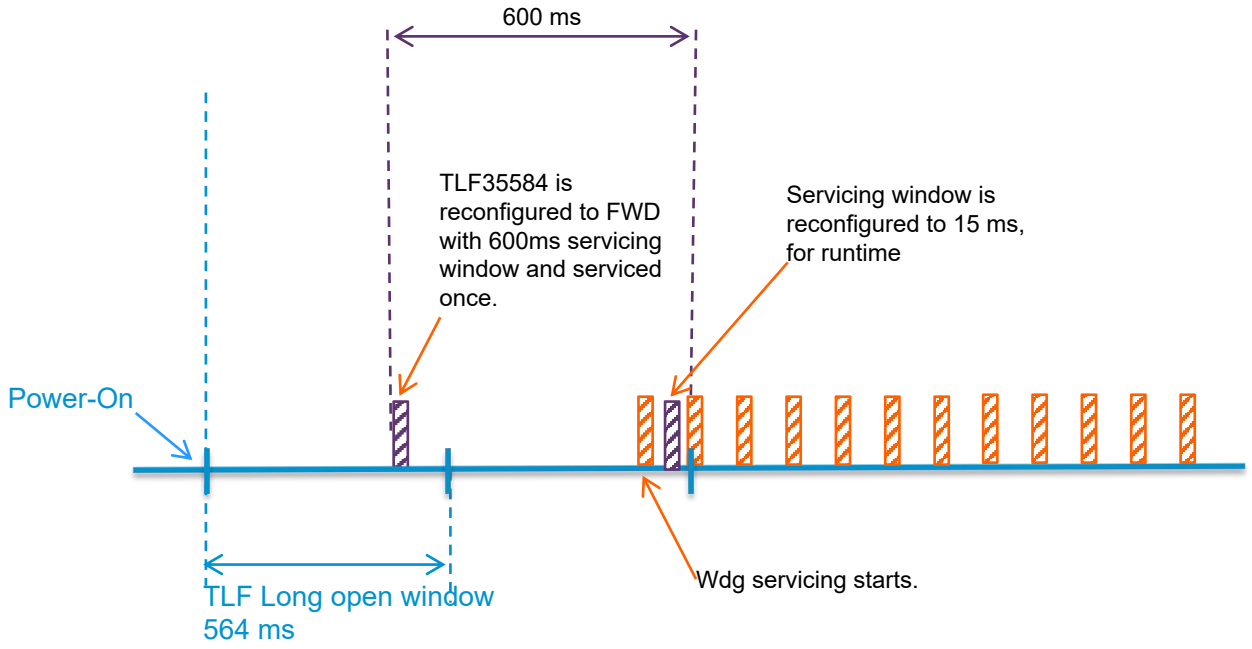
Communication with external watchdog - Q/A Protocol

1. The communication with external watchdog TLF35584 follows a Seed/Signature (Q/A) protocol.
2. Servicing the external watchdog means sending a correct answer (signature) in response to a question (seed).
3. Question - A seed is fetched from the TLF35584.
4. Answer - The result of BIST run-time tests is used to calculate signature, which is sent to TLF35584.
5. The Q/A protocol has inputs from two services:
   - Task Monitoring Service
   - BIST
6. The external watchdog is serviced regularly when no problem is reported.
7. When a problem is reported:
   - by TaskMonitoring - no answer is sent.
   - by BIST - incorrect signature is sent.



Initialization Behaviour

1. In bEIP, the external watchdog TLF35584 is always used in Functional Watchdog mode.
2. However, TLF35584 boots up in Window Watchdog mode. Therefore, at boot-up, the TLF35584 is reconfigured to Functional Watchdog mode.
3. This needs to be done in the Long Open Window of the TLF35584.
4. The functional watchdog is then configured to have a Q/A protocol window of 600ms during the start-up phase.
5. After start-up is complete, the FWD is reconfigured to have a Q/A protocol window of 15ms.



TTTechAuto

03

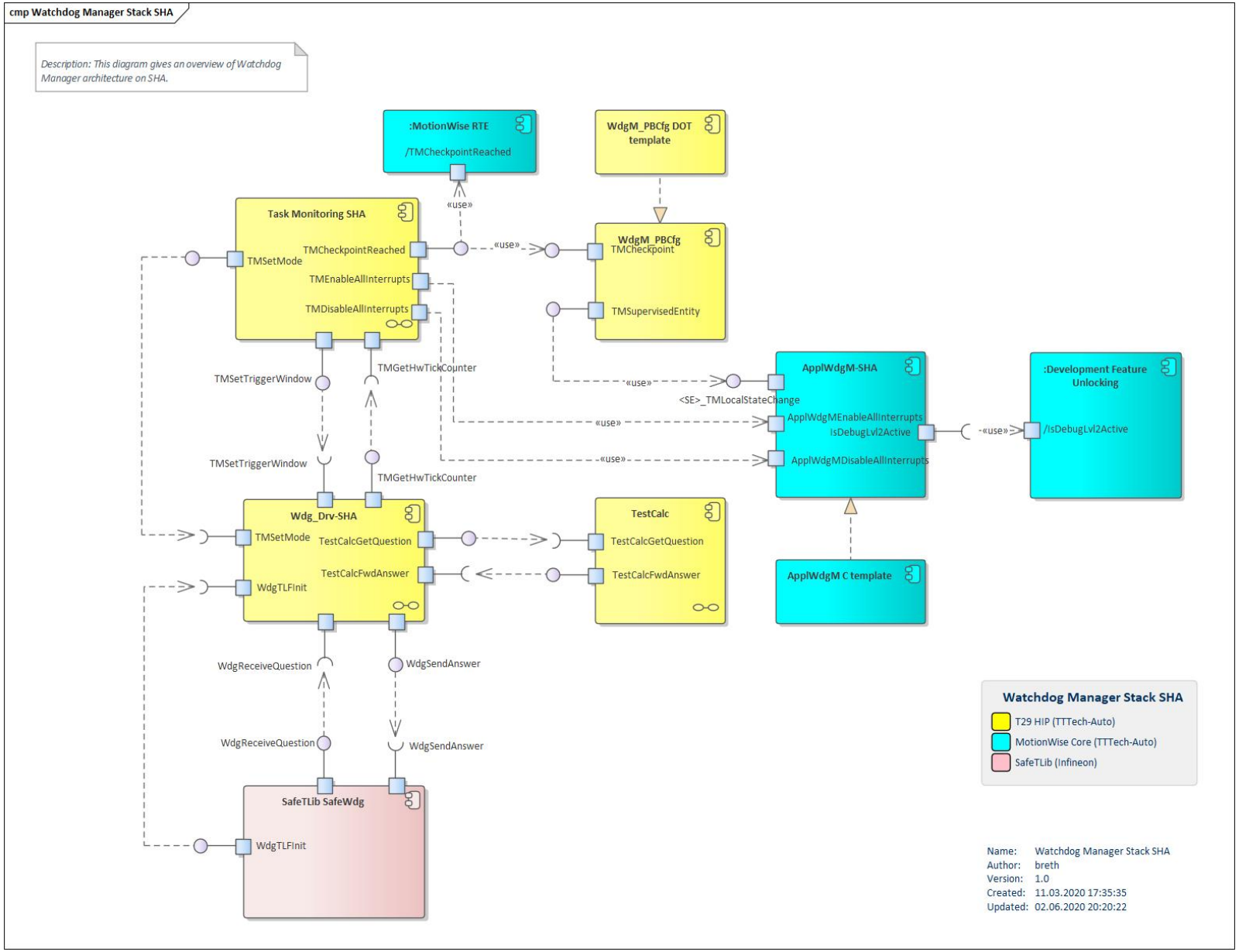
Architecture

Task Monitoring and Watchdog

October 30, 2023

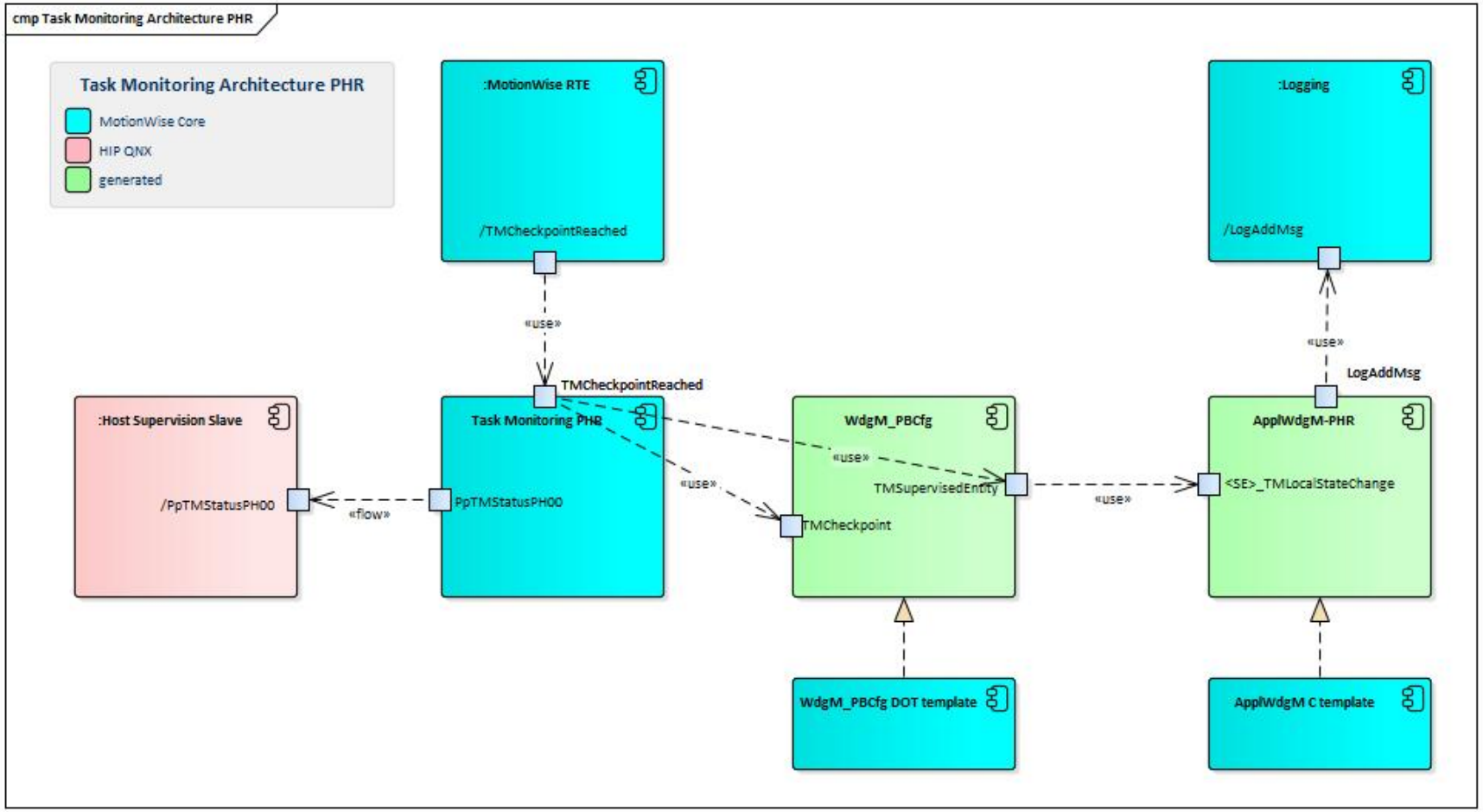
Architecture on SHA

1. Task Monitoring feature uses the Wdg library, which has the following units:
   - Wdg Manager
   - Wdg Interface
   - Wdg Driver
2. Wdg Manager and Wdg Interface are represented in the adjacent diagram by the Task Monitoring SHA block.
3. The Wdg library uses the Infineon's SafeTLib module for communication with the TLF35584.



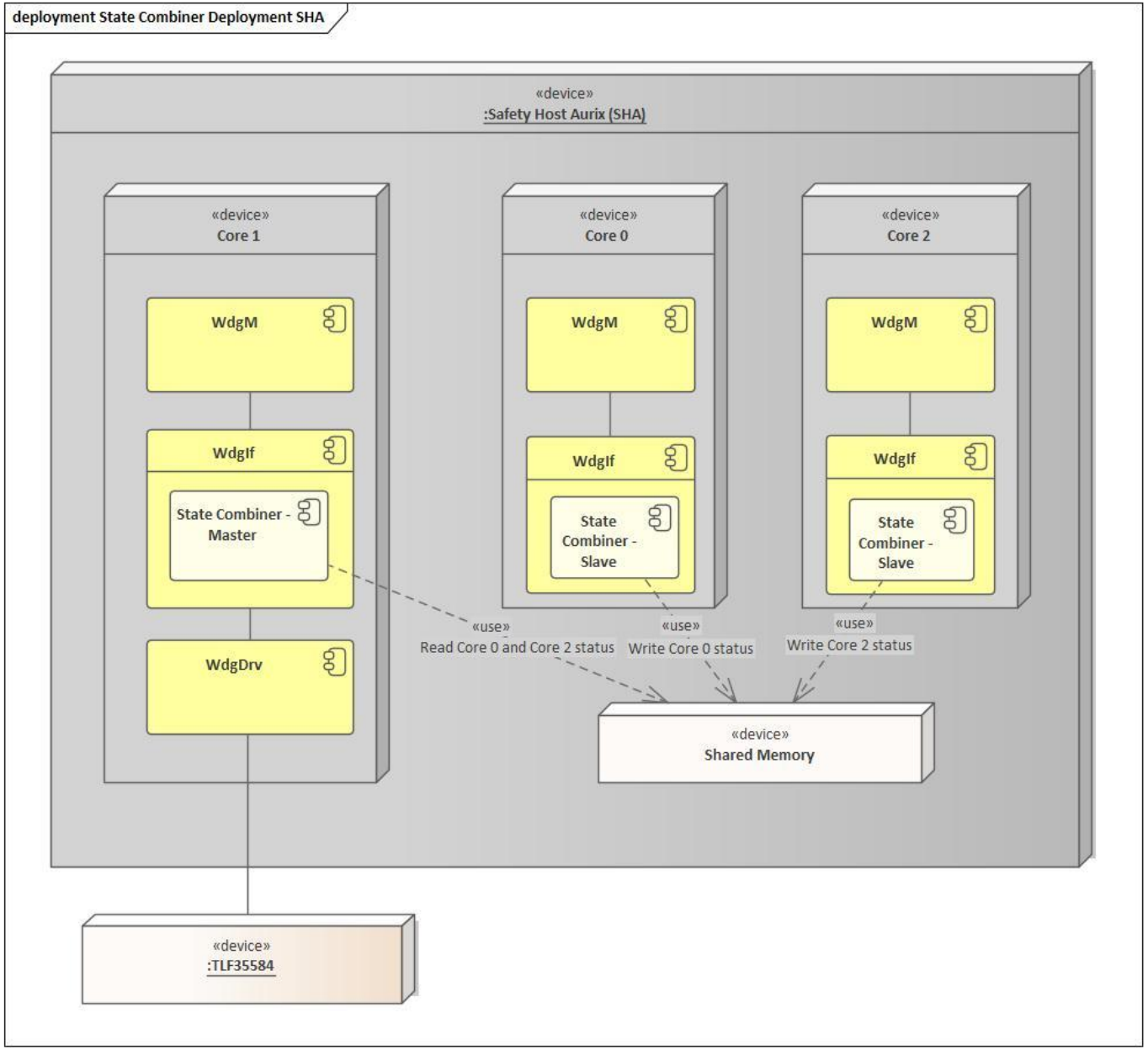
Architecture on PH

1. On the PH, Task Monitoring feature uses the following units:
   - Wdg Manager
   - Wdg Interface
2. Wdg Interface reports the status of Task Monitoring to Host Supervision Slave.



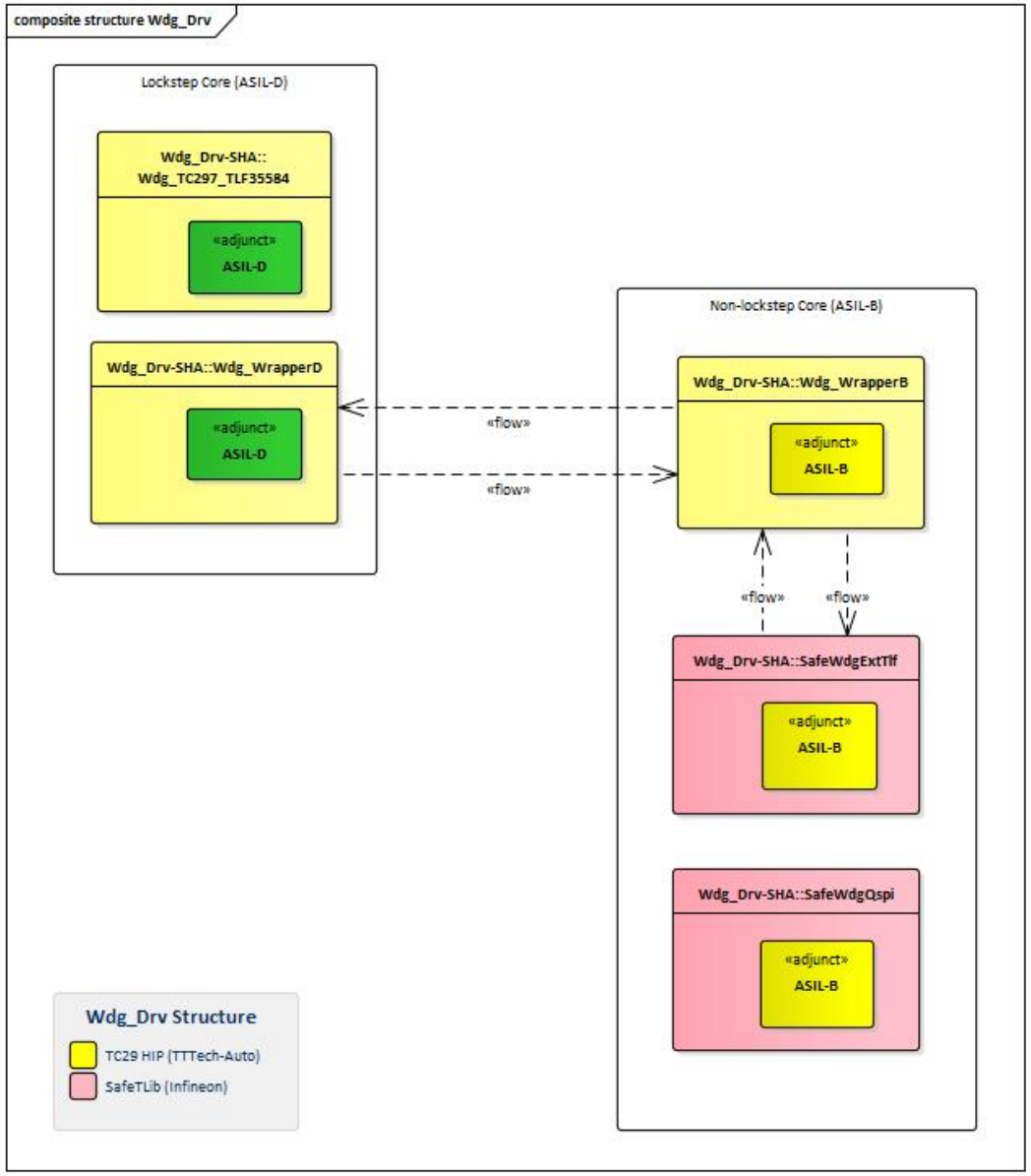
Architecture on SH – State Combiner

1. On the Safety Host, only Core 1 triggers the external watchdog TLF35584.
2. Task Monitoring status is evaluated on each Core independently by separate instances of WdgM.
3. The result from all cores is then combined, and the external watchdog is triggered only if all instances of WdgM report an OK status.
4. This is achieved by the State Combiner functionality.
5. StateCombiner–Master runs on Core 1, and StateCombiner-Slaves run on Core 0 and Core 2.



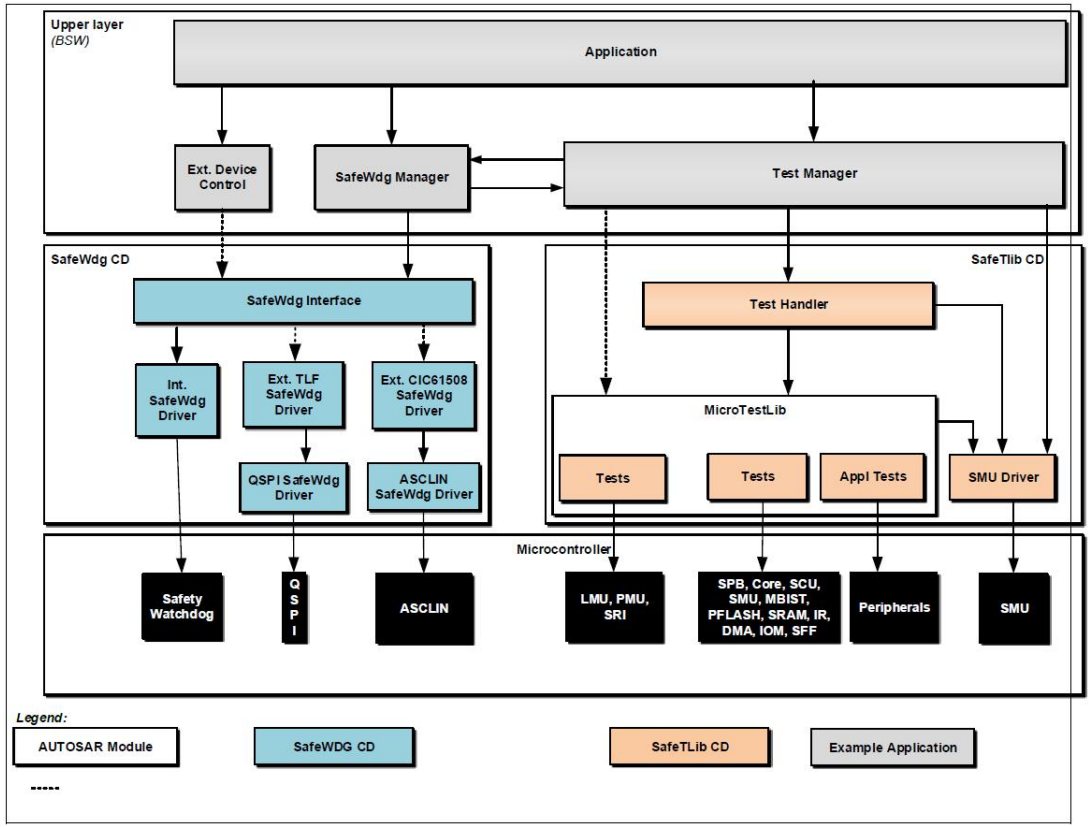
Wdg Drv Architecture

1. Wdg stack runs on the lockstep core.
2. SafeTLib modules run on the non lockstep core. Hence, Wrapper D and Wrapper B are needed as wrappers between the Wdg Drv and the SafeTLib modules.



SafeTLib

1. SafeTlib is a library provided by Infineon which has support for watchdog drivers and microcontroller tests.
2. It is developed as SEooC in ASIL-B quality.
3. It has in-built drivers for external watchdog (specifically TLF35584) and QSPI.



Source: **Aurix SafeTlib User Manual v10.7, 2017-05**

Watchdog Concepts

Logical monitoring and temporal monitoring need to be performed. The options to achieve them are:

1. **External question/answer watchdog**: The logical and the temporal supervision are performed by an external watchdog device.
   - This is achieved by using TLF35584 in *FWD* mode + Task Monitoring service.
2. **Internal safety watchdog + external time-window watchdog**: The logical supervision is performed by the internal safety watchdog and the temporal supervision is performed by an external time-window watchdog.
   - This is achieved by using internal Safety Watchdog + TLF35584 in *WWD* mode.

TTTechAuto

04

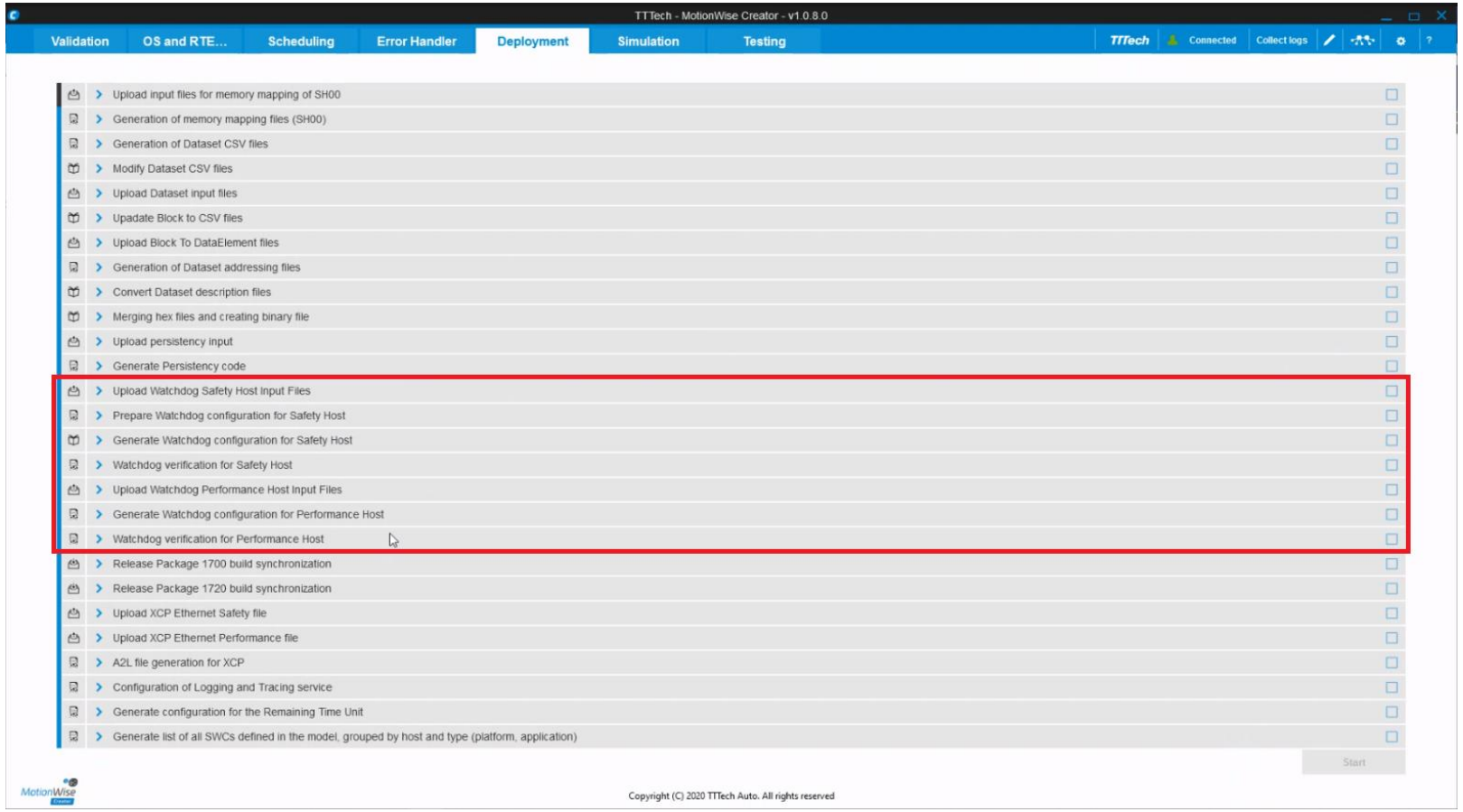
Configuration

Task Monitoring and Watchdog

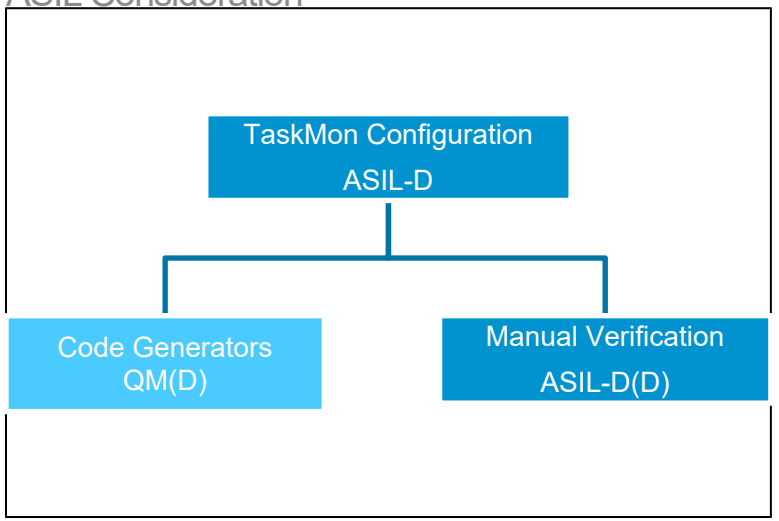
October 30, 2023

Generation of config files

1. The input file for all configurations is **dot_file.template**, which can be edited in text mode.
2. The compile time and post-build configuration files are then generated by executing the following steps from MotionWise Creator.
3. The verification step in the MotionWise Creator only verifies a part of the configuration.
4. Complete verification should be done by manual verification checks, as specified in the bEIP Safety and Integration Manual.



ASIL Consideration



Configurable Parameters - WdgM

TTTechAuto

Parameter	Multiplicity	Range	Type	Default	Config Level	Description
WdgMExpiredSupervisionCycleTol	1	0 ... uint16_max	uint16	0	PIE	This parameter shall be used to define a value that fixes the amount of expired supervision cycles for how long the blocking of watchdog triggering shall be postponed after the global supervision status has reached the state expired.
WdgMFailedSupervisionRefCycleTol	1	0 .. uint16_max	uint16	0	PIE	WdgM post-build time configuration, failed alive supervision cycle tolerance.
WdgMFailedDeadlineRefCycleTol	1	0 .. uint16_max	uint16	0	PIE	Number of tolerated deadline reference cycles in local status FAILED
WdgMFailedProgramFlowRefCycleTol	1	0 .. uint16_max	uint16	0	PIE	Number of tolerated program flow reference cycles in local status FAILED
use_WCET_for_DLMAX	1	0 ,, 1	uint8	0	PIE	Selector for DeadlineMax parameter. 0 = period, 1 = wcet
WdgMDeadlineFactor	1	0 .. 100	float	1,0	PIE	Additional factor on 'wcet', for the WdgMDeadlineMax parameter.
WdgMDevErrorDetect	1	true/false	boolean	FALSE	PIE	Pre-processor switch for enabling the development error reporting.
WdgMDemSupervisionReport	1	true/false	boolean	FALSE	PIE	Pre-processor switch to enable/disable the call to DEM.
WdgMDemReport	1	true/false	boolean	FALSE	PIE	Pre-processor switch to enable/disable calls to DEM in case of production error detection.
WdgMDemReport	1	true/false	boolean	FALSE	PIE	Pre-processor switch to enable/disable calls to DEM in case of production error detection.
WdgMWatchdogMode	1	WDGIF_FAST_MODE/ WDGIF_OFF_MODE/ WDGIF_SLOW_MODE	EcucEnumerationParamDef	WDGIF_SLOW_MODE	PIE	This parameter contains the watchdog mode that shall be used for the referenced watchdog in this Watchdog Manager mode.
WdgMTriggerConditionValue	1	1 .. 65535	int	15	PIE	Value to configure the time window during which the external watchdog accepts triggers from the WdgM Stack. This parameter defines the end of the time window with respect to the last call of the WdgM main function.

Configurable Parameters - WdgIf

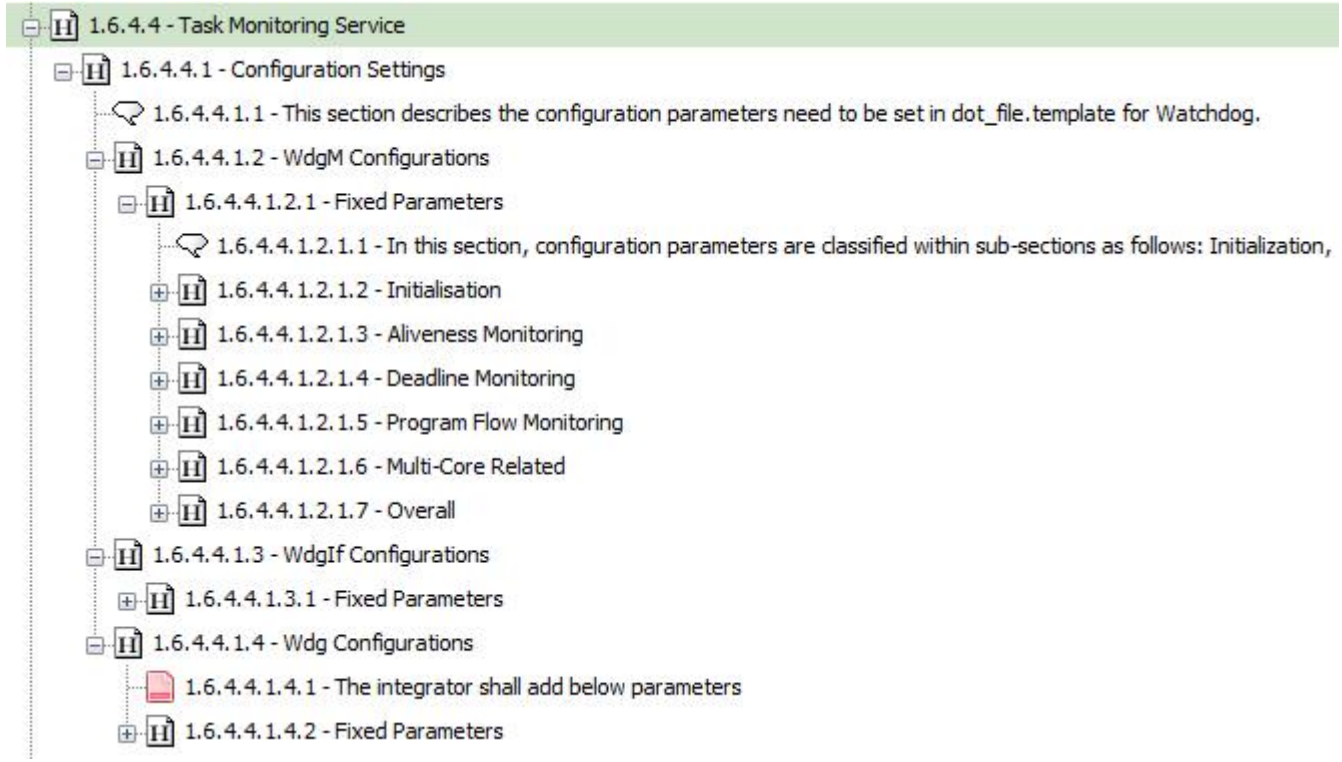
Parameter	Multiplicity	Range	Type	Default	Config Level	Description
WdgIfDevErrorDetect	1	true/false	boolean	FALSE	PIE	Pre-processor switch for enabling the development error reporting.
WdgIfStateCombinerStartUpSyncCycles	1	0 ... 65535	uint16	0	PIE	Number of master cycles during start-up in which the master does not check the slave triggering. Used to synchronize master and slave during start-up.

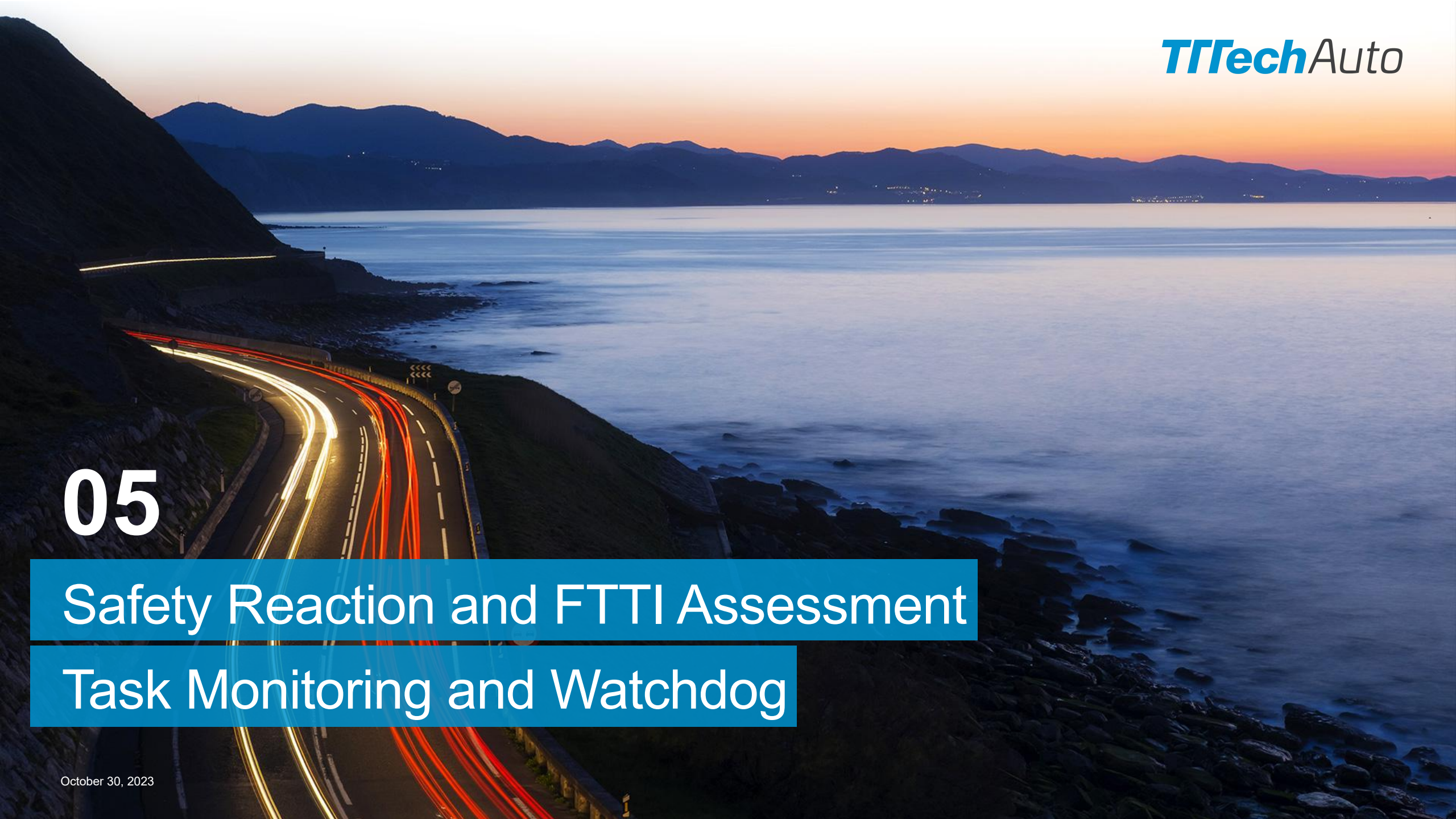
Configurable Parameters – Wdg Drv

Parameter	Multiplicity	Range	Type	Default	Config Level	Description
InitialWindowStart (ms)	1	0 ... uint16_max	uint16	0.0	PIE	Initial minimum time until next watchdog service is allowed.
InitialTimeout (ms)	1	0 ... uint16_max	uint16	200 (ms)	PIE	The initial timeout configures the window on TLF35584 to account for the start-up delay of SH.
ErrorThreshold	1	0 ... 15	uint16	6	PIE	Error threshold defined for TLF to generate reset.
WdgDevErrorDetect	1	true/false	boolean	FALSE	PIE	Pre-processor switch to enable/disable calls to DEM in case of production error detection.
WdgDemReport	1	true/false	boolean	FALSE	PIE	Pre-processor switch to enable/disable calls to DEM in case of production error detection.

Parameters fixed in bEIP

- Fixed parameters are described in the bEIP Software Parameter Document.





TTTechAuto

05

Safety Reaction and FTTI Assessment

Task Monitoring and Watchdog

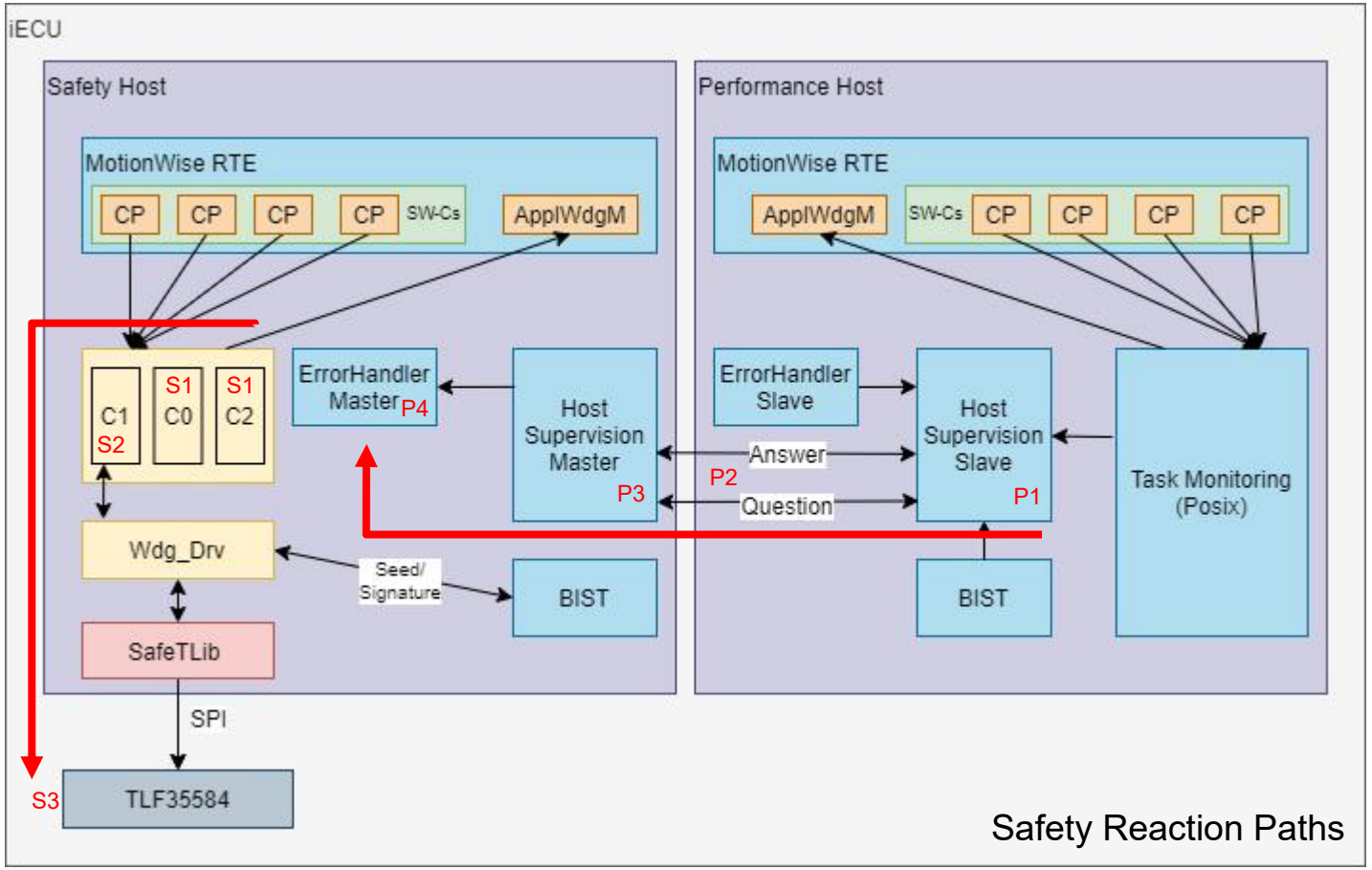
October 30, 2023

Safety Reactions

- Violations detected on the Safety Host trigger the external watchdog TLF35584, which initiates a safety reaction.
- Violations detected on the Performance Host are reported via HSV channel.

FTTI Assessment

- Technical Safety Requirement:
  - FDTI + FRTI < FTTI
  - Assumed FTTI = 100ms
  - FDTI depends on the type of monitoring – Aliveness, Deadline and Program Flow.
  - FRTI depends on the configured safety reaction path.



Safety Reaction Paths

- $FRTI_{PH}$ = (P1) HSVS Slave runnable period +
  (P2) Middleware scheduling period +
  (P3) HSV Reaction Time +
  (P4) Error Handler reaction time

- $FRTI_{SH}$ = (S1) StateCombiner Delay +
  (S2) WdgMExpiredSupervisionCycleTol * WdgMSupervisionCycle +
  (S3) (Error_Threshold / 2) * WdgMTriggerConditionValue

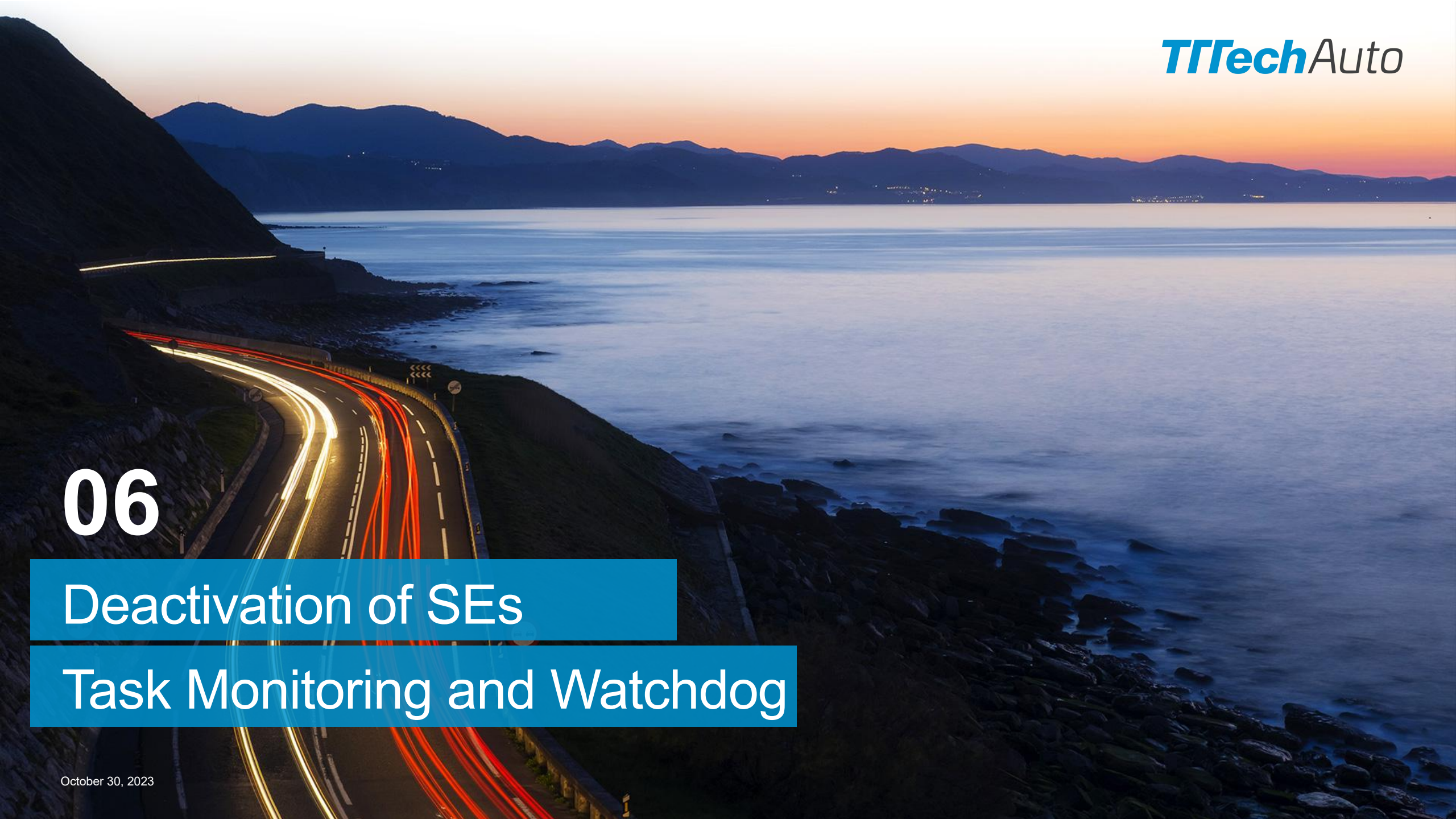
StateCombiner Delay = 0, for all Core 1, as it is the master
StateCombiner Delay = 20 ms, for Core 0 and Core 2,
because master check slave triggers every alternate cycle.

Quick Safety Reaction Path - ApplWdgM

TTTechAuto

1. On every LocalState change, a callback function in ApplWdgM.c is invoked.
2. This callback can be used to log information and to trigger a safety reaction, thereby minimizing the safety reaction path.
3. FRTI is reduced to 0.
4. In bEIP, when the LocalState changes to EXPIRED, the callbacks trigger a safety reaction by shutting the CAN Tx pins via the Port Emergency Shutdown feature of the Aurix.

```
void R_SH00_MiddlewareQM_9_WdgM_LocalStateChange_Cbk(WdgM_LocalStatusType Status)
{
    switch ( Status)
    {
        case WDGM_LOCAL_STATUS_FAILED:
        {
            if ( ApplWdgM_CurrentCycleMesswert.WdgM_V_SEID == NO_SE_ID )
            {
                ApplWdgM_CurrentCycleMesswert.WdgM_V_SEID = (WdgM_SupervisedEntityIdType)SE_R_SH00_MiddlewareQM_9;
            }

            Log_AddMsg (SWCID_CtCdMiddlewareQM_SH00, LOG_GROUP_8, LOG_WARNING, " R_SH00_MiddlewareQM_9 execution flow violation!");

            break;
        }

        case WDGM_LOCAL_STATUS_EXPIRED:
        {
            if ( IsDebugModeLevel2Active() == (boolean)FALSE )
            {
                #ifndef TASKMON_DISABLE_WDG_REACTION
                (void)EHM_SET_Aurix_Can_Pins_Emergencymode();
                #endif
            }

            if ( ApplWdgM_CurrentCycleMesswert.WdgM_ER_SEID == NO_SE_ID )
            {
                ApplWdgM_CurrentCycleMesswert.WdgM_ER_SEID = (WdgM_SupervisedEntityIdType)SE_R_SH00_MiddlewareQM_9;
            }

            Log_AddMsg (SWCID_CtCdMiddlewareQM_SH00, LOG_GROUP_8, LOG_ERROR, " R_SH00_MiddlewareQM_9 execution flow ERROR!");

            if ( HSVM_IsTLFWatchdogDisabled() == FALSE )
            {
                uint8 NewResetReason = (uint8)RSTRSN_WDGM_RESERVED_START + (uint8)SE_R_SH00_MiddlewareQM_9;
                /* PRQA S 2991 2*/  /* PRQA S 2995 2*/
                /* PRQA S 4342 3 */ /* The cast is ok because the result of the cast will be checked to be within the correct range of the enum before assignment */
                if (( (e_SRR_RReason)NewResetReason >= RSTRSN_WDGM_RESERVED_START ) && ( (e_SRR_RReason)NewResetReason <= RSTRSN_WDGM_RESERVED_STOP )) /* PRQA S 2991
                {
                    SRR_SetResetReason( (e_SRR_RReason)NewResetReason );
                }
            }

            break;
        }

        case WDGM_LOCAL_STATUS_OK:
        {
            if ( (WdgM_SupervisedEntityIdType)SE_R_SH00_MiddlewareQM_9 == ApplWdgM_CurrentCycleMesswert.WdgM_V_SEID )
            {
                ApplWdgM_CurrentCycleMesswert.WdgM_V_SEID = NO_SE_ID;
            }

            break;
        }
```



TTTechAuto

06

Deactivation of SEs

Task Monitoring and Watchdog

October 30, 2023

Deactivation of Supervised Entities

TTTechAuto

- **WdgM_DeactivateSupervisionEntity**

  1. The function marks an entity for deactivation. An entity can only be deactivated when its local state is WDGM_LOCAL_STATUS_OK or WDGM_LOCAL_STATUS_FAILED.
  2. The deactivation itself happens at the end of the supervision cycle inside the WdgM_MainFunction().
  3. When an entity is deactivated then its checkpoints are not evaluated anymore, and the entity local state is WDGM_LOCAL_STATUS_DEACTIVATED.
  4. This function can deactivate a supervised entity only before its initial checkpoint was passed or after its end checkpoint was passed.

  - Notes:
    1. When an entity is deactivated, the global transitions to this entity are not evaluated.
    2. After SE deactivation the function WdgM_GetLocalStatus () can be used to check the SE local state.
    3. Using this function can degrade system safety. The deactivation of entity supervision in safety-related products needs special attention to avoid unintended supervised entity deactivation.

- **WdgM_ActivateSupervisionEntity**

  1. The function marks an entity for activation. An entity can only be activated when its local state is WDGM_LOCAL_STATUS_DEACTIVATED.
  2. The activation itself happens at the end of the supervision cycle inside the WdgM_MainFunction().

  - Notes:
    1. In the same call of WdgM_MainFunction(), first the local states of all supervised entities and the global state are set, then the supervised entity is activated.
    2. After SE activation the function WdgM_GetLocalStatus() can be used to check the SE local state.

TTTechAuto

07

Schedule Outage Check

Task Monitoring and Watchdog

October 30, 2023

Schedule Outage Check

TTTechAuto

- **Who guards the guardians?**

    1. The WdgM_MainFunction monitors the temporal behaviour of all other runnables on each core.
    2. The temporal behavior of WdgM_MainFunction is monitored by an independent check, separately on each core.
    3. Schedule Outage Check -
        1. Time difference between two consecutive executions of WdgM_MainFunction is checked to be within the range of 5ms to 15ms.
        2. Time difference is evaluated using **STM Timer0** of respective core.
        3. In case of violation, error **ERR_OS_SH00_SCHEDULE_CYCLE_OUTAGE** is reported to the Error Handler.

TTTechAuto

08

Exclusions

Task Monitoring and Watchdog

October 30, 2023

Exclusions

TTTechAuto

The following entities are excluded from the purview of Task Monitoring Service:

- BSW Tasks
- Free Running Apps

BSW Task Name	Safety Level	Functions Invoked	Rationale for no Task Monitoring
Task_10ms_QM_C0	QM	IoHwAb_LSHS_Setup_Async CDD_Uart_Task	QM functionality.
Task_10ms_QM_C2	QM	SwtMng_MainFunction XcpAppl_KvsEvent Xcp_Event	QM functionality.
Task_10ms_ASILB_C2	ASIL-B	WdgM_MainFunction	This function is monitored by schedule outage check.
Task_Log	QM	PerfMeas_TraceStackConsumption	QM functionality.
Task_10ms_ASILB_C0	ASIL-B	WdgM_MainFunction	This function is monitored by schedule outage check.
Task_10ms_ASILD_C1	ASIL-D	WdgM_MainFunction StbM_MainFunction Wdg_WrapperD_ClearRequest	The entire task is monitored by a schedule outage check.
Task_10ms_ASILB_C2_TstM	ASIL-B	TstM_TLF_CyclicHealthTest TstM_STL_RunSFRTests Wdg_WrapperB_CommandInvoke	The TLF35584 is indirectly monitoring the execution of this task.
Task_10ms_ASILB_C2_WrBMain	ASIL-B	Wdg_WrapperB_MainFunction Wdg_WrapperB_FetchSeed	The TLF35584 is indirectly monitoring the execution of this task.
Task_Sync_C0 **Task_Sync_C1** **Task_Sync_C2**	QM	TSch_SyncScheduleTables	QM functionality.
Task_BSW_C0 **Task_BSW_C2**	QM	_MainFunctions of all BSW modules.	QM functionality.

TTTechAuto

HQ, Vienna, Austria

-auto.com | -auto.com | +43 1 585 65 38-5000

TTTechAuto

Time for Q&A

Srikrishna-Balaji Prayaga

Software Architect
Srikrishna-Balaji.Prayaga@tttech-auto.com

Follow us on **LinkedIn**
www.tttech-auto.com