

Specyfikacje techniczne

A. Część ogólna

Przedmiotem zamówienia jest wykonanie rozwoju oprogramowania integrującego urządzenia w systemie sterowania oraz oprogramowania systemu archiwizacji danych kontrolno-pomiarowych FAIR”. Zakres usługi ma obejmować integrację sprzętu i systemu sterowania oraz oprogramowania bazy danych pomiarowych do sterowania dla akceleratora SIS100 w ramach polskiego wkładu rzeczowego do FAIR tj. dwóch pakietów roboczych, WP1 i WP2. Wykonawca ponosi odpowiedzialność za opracowanie, przetestowanie, dostarczenie i wdrożenie oprogramowania. Wykonawca jest odpowiedzialny za dostarczenie informacji technicznych dotyczących dostarczenia oprogramowania, a także za zapewnienie personelu niezbędnego do wdrożenia i testowania oprogramowania.

1. WP1 - FESA (architektura oprogramowania front-endu)

Wykonawca opracuje nowe klasy FESA (FRONT-END SOFTWARE ARCHITECTURE) i zmigruje istniejące klasy FESA niektórych urządzeń we współpracy z zespołem GSI ACO i grupami technicznymi odpowiedzialnymi za urządzenia używane w aplikacjach GSI i przyszłych aplikacjach FAIR. Pakiet Pracy obejmuje w szczególności:

- Analizę aktualnych klas urządzeń
- Planowanie migracji do FESA
- Projektowanie i wdrażanie klas FESA.
- Dokumentacja wykonanych prac wraz z projektem i opisem API (Application Programming Interface).

Pakiet Roboczy składa się z pięciu etapów:

- Zaprojektowanie i wdrożenie standardowej klasy FESA dla nieskomplikowanych urządzeń, wdrożenie w całym systemie dla FAIR
- Refaktoryzacja i modularyzacja wtyczki Eclipse używanej do projektowania i rozwoju klasy FESA, w celu zwiększenia elastyczności na przyszłe zmiany.
- Tworzenie prototypów do podłączenia urządzeń i interfejsów do eksperymentów oraz systemów infrastruktury akceleratora do systemu sterowania z wykorzystaniem protokołów komunikacyjnych OPC/UA i MQTT.
- Zapewnienie wsparcia YOCTO (systemu typu open-source implementacji w systemie LINUX niezależnym od platformy sprzętowej) jako standardowego systemu wykonawczego FAIR i platformy systemu operacyjnego czasu rzeczywistego, w tym YOCTO Software Development Kit (SDK) i środowiska dla wbudowanego FESA.
- Wspieranie procesów integracji, wdrażania i testowania oprogramowania i klas FESA na wbudowanych kontrolerach. Integracja repozytorium Git z potokiem CI/CD.

Najważniejsze kamienie milowe przedstawiono w tabeli 1. Oczekuje się osiągnięcia kamieni milowych zgodnie z zapisanym harmonogramem . Można je zmienić jedynie za obopólną zgodą na piśmie. Spełnienie Kryteriów Walidacji zostanie sprawdzone przez FAIR GmbH.

2. WP2 - System Archiwizacji ze szczególnym uwzględnieniem zastosowań oprzyrządowania wiązki dla FAIR

System archiwizacji musi zapewniać pełną informację o ustawieniach akceleratora SIS100 (ustawień magnesów, próżni etc) oraz stanu systemu sterowania akceleratorem, umożliwiającą jednoznaczna analizę i interpretację. System powinien zaspokajać potrzeby operatorów, działów

technicznych i osób współpracujących w ramach eksperymentów. System archiwizujący powinien mieć możliwość odtworzenia trybu pracy wszystkich elementów akceleratora w dowolnym momencie oraz śledzenia zmian trybów pracy i rozkładu wiązek w wybranym okresie.

System archiwizacji będzie zawierał Standardowe i Zaawansowane procedury analizy.

Standardowe procedury:

- Protokoły zmian dla łańcucha produkcyjnego wiązki lub sekcji maszyn
- Protokoły czasu wiązki
- Parametry wiązki na tarczach i miejscach kończących bieg wiązki (beam dump)
- Kontrola systemu technicznego
- Wykresy stabilności wiązki dla eksperymentatorów i operatorów
- Monitorowanie trendów w ramach kontroli granic tolerancji

Zaawansowane procedury:

- Szczegółowa analiza
- Wydobywanie informacji poza danymi detektora
- Informacje o stanie w momencie awarii
- Rejestrowanie informacji diagnostycznych i operacyjnych
- Dane i status ustawień akceleratora.

3. Plan kamieni milowych dla przedmiotu zamówienia

#	Kod aktywności	Opis zadania	Kryteria walidacji	Uzgodnione daty dostaw w niniejszym aneksie liczone od momentu podpisania umowy
0	M4		Podpisanie umowy	0
1	M6	Kwartalne przeglądy projektu i uwag odbiorcy	PDR (Wstępny Przegląd Projektu) zaakceptowany	Każdy kwartał
2	M6b	Do testowania obu zadań (WP1-2) zostanie wykorzystany czas wiązki 2024	FDR gotowy dla obydwu WP	15 miesięcy
3	M7	Końcowy Przegląd Projektu (FDR)	FDR zaakceptowany dla obydwu WP	15 miesięcy
4	S006.M10	Dostawa pełnej dokumentacji	Wszystkie elementy umowy dostarczone	21 miesięcy

Dodatkowy podział w ramach wykonania w.w etapów:

WP1	FESA	Ilość (części)	Kryteria walidacji	Data wykonania
C1.1.	Projektowanie i wdrażanie Generic klas FESA wraz z podklasami.	13	Protokół akceptacyjny po każdej części wykonanej osobno	

C1.2	Refaktoryzacja i modularyzacja wtyczki Eclipse dla FESA	1	Protokół akceptacyjny	
C1.3	Prototypy do integracji automatyki przemysłowej i protokoły M2M		Protokół akceptacyjny	
C1.4	YOCTO Wsparcie	1	Protokół akceptacyjny	
C1.5	CI/CD Wsparcie	1	Protokół akceptacyjny	
WP2	System Archiwizacji	Ilość (części)	Kryteria walidacji	Data wykonania
C.2	Uaktualnienie ogólnej stabilności i rzetelności	1	Protokół akceptacyjny	
C2.2	C2.2 Uaktualnienie "Node Orchestration" Konfiguracji	1	Protokół akceptacyjny	
C2.3	C2.3 Uaktualnienie Administracji GUI	1	Protokół akceptacyjny	

B. Część szczegółowa

Spis treści

1.	Cel i klasyfikacja dokumentu	
1.1.	Wprowadzenie do FAIR i sterowania akceleratora.....	
2.	Skróty, terminy i definicje	
3.	Zakres systemu technicznego.....	
3.1.	Przegląd systemu	
3.2.	Zakres dostawy.....	
3.3.	Nakład pracy	
3.3.1.	Obecność na kampusie FAIR/GSI.....	
3.3.2.	Pomiary serwisowe i testowe na miejscu	
3.3.3.	Rozwój oprogramowania FEC	
3.3.3.1.	Projekt i implementacja ogólnej klasy FESA z podklasami (WP C1.1).....	
3.3.3.2.	Refaktoryzacja i modularyzacja wtyczki Eclipse do FESA (WP C1.2)	
3.3.3.3.	Prototypy do integracji automatyki przemysłowej i protokoły M2M (WP C1.3).....	
3.3.3.4.	Wsparcie YOCTO (WP C1.4).....	
3.3.3.5.	Wsparcie CI/CD (WP C1.5).....	
3.3.4.	System archiwizacji	
3.3.4.1.	Ulepszenia ogólnej stabilności i niezawodności (WP C2.1)	
3.3.4.2.	Ulepszenia orkiestracji i konfiguracji węzła (WP C2.2)	
3.3.4.3.	Ulepszenia zarządzania GUI (WP C2.3)	
3.4.	Części zamienne.....	
4.	Specyfikacja systemu.....	
4.1.	Wymagania techniczne dotyczące komponentów systemu	
4.1.1.	Rozwój oprogramowania systemu sterowania	
4.1.2.	Środowisko rozwoju oprogramowania	
4.2.	Wymagania funkcjonalne (lista parametrów).....	
5.	Procedura.....	
5.1.	Faza projektowania.....	
5.2.	Faza produkcji.....	

6.	Zapewnienie jakości, testy i akceptacja	
6.1.	System zapewnienia jakości wykonawcy	
6.2.	Fabryczny test akceptacyjny (FAT)	
6.3.	Test akceptacyjny na miejscu (SAT)	
7.	Dokumentacja	
8.	Dokumenty powiązane	

1. Cel i klasyfikacja dokumentu

Dokument ten stanowi obowiązkowy jako opis techniczny dla zakresu objętego przedmiotem niniejszego zamówienia

Zakres zastosowania tego dokumentu obejmuje podzbiór urządzeń sterujących akceleratora dla FAIR i istniejących działań modernizacyjnych GSI, związanych z integracją sprzętu i systemu, a także oprogramowaniem bazy danych pomiarowych.

Krótkie opisy projektu FAIR i sterowania akceleratora znajdują się w poniższej sekcji.

1.1. Wprowadzenie do FAIR i sterowania akceleratora

Obecnie w Darmstadt w Niemczech budowany jest międzynarodowy ośrodek akceleratorowy FAIR, jeden z największych projektów badawczych na świecie. W FAIR materia, która zwykle istnieje tylko w przestrzeni kosmicznej, zostanie wyprodukowana w laboratorium do badań. Naukowcy z całego świata będą mogli uzyskać nowy wgląd w strukturę materii i ewolucję wszechświata od Wielkiego Wybuchu do chwili obecnej. FAIR będzie jednym z największych i najbardziej złożonych systemów akceleratorów na świecie. Akcelerator FAIR będzie miał unikalną zdolność dostarczania wiązek cząstek wszystkich pierwiastków chemicznych (lub ich jonów), a także antyprotonów. Cząstki zostaną przyspieszone do prędkości bliskiej prędkości światła w akceleratorach FAIR i udostępnione do eksperymentów naukowych. W FAIR będą generowane wiązki cząstek o niespotykanej wcześniej intensywności i jakości. Sercem obiektu jest synchrotron SIS100 o obwodzie 1100 metrów umieszczony pod ziemią. Istnieją również dodatkowe pierścienie eksperymentalne i stacje eksperymentalne o łącznej długości kilku kilometrów linii wiązek. Istniejące akceleratory GSI Helmholtzzentrum für Schwerionenforschung będą służyć jako iniektor dla nowego obiektu FAIR (SIS100). Więcej informacji na temat projektu FAIR można znaleźć na stronie internetowej projektu: <https://www.gsi.de/en/researchaccelerators/fair>.

Sterowanie akceleratorem jest centralnym podsystemem kontroli łańcucha akceleratorów GSI i FAIR. Zadaniem działu sterowania akceleratora jest zapewnienie infrastruktury technicznej i rozwiązań programowych dla personelu operacyjnego akceleratora, a także ekspertów technicznych w zakresie wszystkich podsystemów do ustawiania, optymalizacji, monitorowania maszyn akceleratora, a także wszystkich podsystemów technicznych, z których składa się kompleks akceleratora. Dodatkowo system sterowania zapewnia systemy ochrony maszyn i personelu. Wszystkie komponenty opisane w niniejszej specyfikacji są częścią wyposażenia systemu sterowania i usług dla FAIR.

2. Skróty, terminy i definicje

ADC	Przetwornik analogowo-cyfrowy (Analog-to-Digital Converter)
ACS	System sterowania akceleratora (Accelerator Control System)
ACO	Dział sterowania akceleratora (Accelerator Controls Department)
BD	Diagnostyka wiązki (Beam Diagnostics)
BEA	Dział oprzyrządowania wiązki (BEAm instrumentation department)
BPM	Nadzór położenia wiązki (Beam Position Monitor)
CDR	Raport z projektu koncepcyjnego (Conceptual Design Report)
CID	Numer identyfikacyjny komponentu (Component Identification Number)
COFB	Informacje zwrotne z zamkniętej orbity Closed-Orbit Feedback)
CPU	Jednostka centralna procesora (Central Processing Unit)
DAQ	Pozyskiwanie danych (Data Acquisition)
FAIR	Ośrodek badań nad antyprotonami i jonami Facility for Antiproton and Ion Research
FAT	Fabryczny test akceptacyjny (Factory Acceptance Test)
FDR	Końcowy przegląd projektu (Final Design Review)
FEC	Sterownik front-end (Front-End Controller)
FESA	Architektura oprogramowania front-end (Front-End Software Architecture)
FPGA	Programowalne macierze bramek (Field Programmable Gate Array)
FTRN	Węzeł odbiorczy czasu FAIR (FAIR Timing Receiver Node)
GMT	Ogólny system pomiaru czasu maszyny (General Machine Timing System)
GSI	GSI Helmholtzzentrum für Schwerionenforschung GmbH
HEBT	Transport wiązki wysokoenergetycznej (High Energy Beam Transport)
IPMI	Inteligentny interfejs zarządzania platformą (Intelligent Platform Management Interface)
LAN	Sieć lokalna (Local Area Network)
LSA	Architektura oprogramowania LHC (LHC Software Architecture)
pTCA	Architektura mikrokomputerów telekomunik. (Micro Telecom Computing Architecture)
NIM	Moduł oprzyrządowania jądrowego (Nuclear Instrumentation Module)
PLC	Programowalny sterownik logiczny (Programmable Logic Controller)
p-Linac	Proton Linac (akcelerator liniowy)
PXE	Środowisko wstępnego uruchomienia komputera (Preboot Execution Environment)
SAT	Test akceptacyjny na miejscu (Site Acceptance Test)
SMA	Adapter przykręcany (Screw Mounted Adapter)
SCR	Ekran scyntylacyjny (Scintillating Screen)
UHV	Bardzo wysoka próżnia (Ultra-High Vacuum)
VSWR	Współczynnik fali stojącej napięcia (Voltage Standing Wave Ratio)
XHV	Ekstremalnie wysoka próżnia (Extreme High Vacuum)

3. Zakres systemu technicznego

3.1. Przegląd systemu

Wspólna specyfikacja systemu sterowania akceleratorem [2] zawiera przegląd architektury oprogramowania i układu systemu sterowania FAIR. Ogólnie rzecz biorąc, wszystkie systemy można

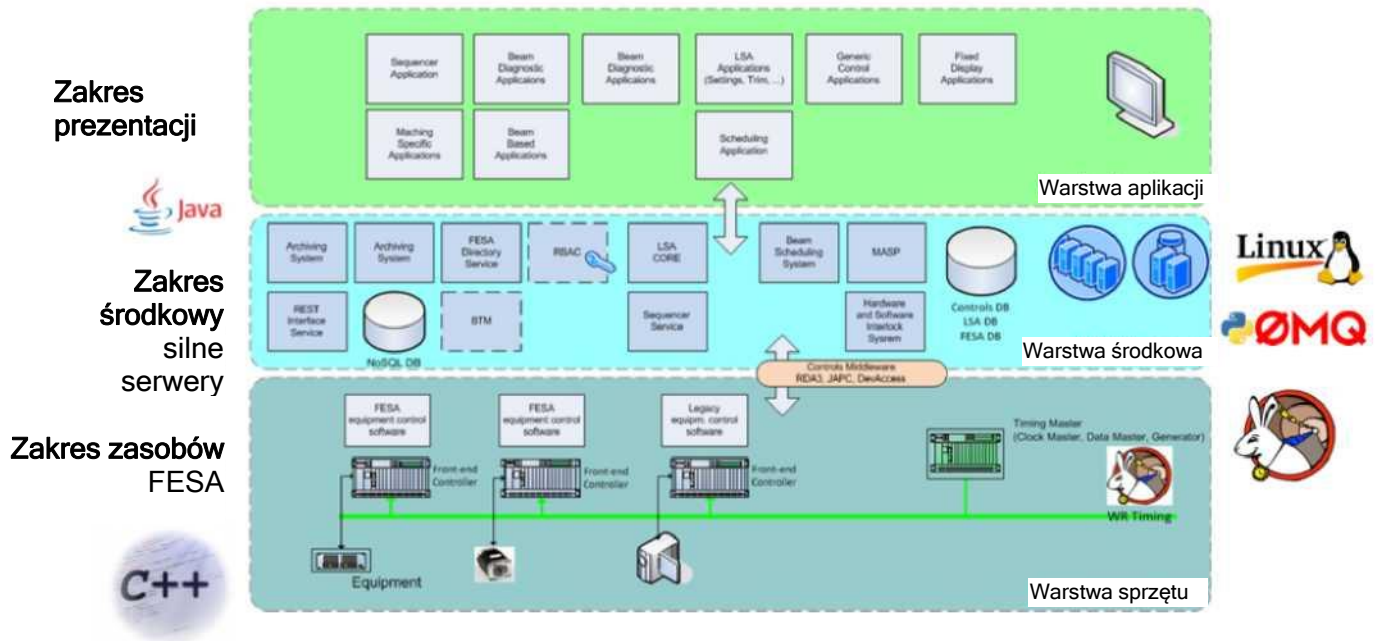
podzielić na 3 warstwy lub poziomy, jak pokazano na rysunku 1.

1. Warstwa zasobów z wbudowanymi sterownikami (sterowniki front-end, FEC) do łączenia i sterowania fizycznym sprzętem akceleratora. Warstwa ta obejmuje precyzyjny system dystrybucji czasu i zdarzeń. Systemy w tej warstwie są w większości sterowane w czasie rzeczywistym i składają się z określonego sprzętu, w większości opracowanego wewnątrz z kodem FPGA. Oprogramowanie jest tworzone w języku C++ przy użyciu środowiska FESA (klasy FESA). FESA (Front End Software Architecture) to niskopoziomowa struktura oprogramowania do sterowania i pozyskiwania danych, pierwotnie opracowana w CERN. Służy do integracji różnych urządzeń z systemem sterowania. Ideą oprogramowania FESA jest uruchamianie kodu urządzenia jak najbliżej urządzenia.
2. Warstwa środkowa z serwerami, które zapewniają ogólne usługi dla warstwy zasobów i warstwy prezentacji, w tym systemy pośredniego oprogramowania komunikacyjnego. Warstwa ta obejmuje niezbędne usługi, takie jak bazy danych, usługi pośredniczące w komunikacji, usługi blokady, a zwłaszcza system zarządzania ustawieniami akceleratora LSA.
3. Warstwa aplikacji to warstwa techniczna z bezpośrednim interfejsem do zespołu operacyjnego akceleratora i specjalistów akceleratora. Duża liczba wyspecjalizowanych programów z graficznym interfejsem użytkownika umożliwia planowanie, ustawianie, dostrajanie, monitorowanie i optymalizację ustawień łańcucha akceleratorów GSI/FAIR. Programy aplikacyjne są zazwyczaj implementowane w JavaFX.

General Machine Timing System to centralny system, który ściśle synchronizuje długi łańcuch akceleratorów. Składa się on z rozproszonego systemu generowania zdarzeń opartego o domenie czasu. Synchronizacja czasu jest osiągnięta za pomocą White Rabbit (WR) [27], w pełni deterministycznej magistrali polowej opartej na sieci Ethernet do przesyłania i synchronizacji zegara. Kluczowymi komponentami GMT jest tak zwany Data Master (DM), który planuje działania poprzez nadawanie komunikatów, sieć WR i węzły Timing Receiver (TR) [18] wykonujące odpowiednie działania maszynowe w czasie rzeczywistym.

Oprogramowanie front-end jest również zdolne do odbierania i reagowania na sygnały czasowe, dystrybuowane przez optyczną sieć White Rabbit i dostępne przez oprogramowanie wykorzystujące bibliotekę SAFT (Simplified API For Timing).

Ponadto architektura systemu obejmuje przemysłowy system sterowania dla powolnych aplikacji sterujących, w szczególności dla niektórych systemów infrastruktury technicznej akceleratora, takich jak system kontroli próżni dla wszystkich maszyn i linii wiązek, system sterowania instalacji kriogenicznych ciekłego helu (LHe), system dystrybucji, lokalna dystrybucja do nadprzewodzących układów zbiorczych i magnesów oraz system próżni izolacyjnej. Przemysłowy system sterowania jest również wykorzystywany w systemie ochrony maszyn systemów sterowania i systemie kontroli dostępu personelu, który będzie chronił użytkowników przed szkodami spowodowanymi promieniowaniem i innymi zagrożeniami poprzez kontrolę dostępu do monitorowanych obszarów i zapewnienie odpowiednich funkcji bezpieczeństwa, które wynikają z formalnej analizy ryzyka.



Rysunek 1. Schematyczny przegląd ogólnej architektury systemu sterowania

Zakres rezultatów obejmuje elementy we wszystkich trzech warstwach architektury ogólnej. Jego celem jest uzupełnienie lub funkcjonalne rozszerzenie istniejących rozwiązań, zapewnienie rozwiązań umożliwiających modernizację istniejącego łańcucha akceleratorów działającego jako iniektor FAIR do standardów systemu sterowania FAIR oraz obsługa bieżącego programu „Fazy 0 eksperymentu FAIR”. Rezultaty zostaną podzielone na odpowiednio zdefiniowane pakiety robocze w poniższej sekcji.

3.2. Przedmiot zamówienia

Poniższa lista opisuje zakres usług do Wykonania systemu sterowania i wsparcia inżynierskiego. Podane nakłady pracy na rozwój dla poszczególnych pakietów prac zostały oszacowane przez specjalistów z FAIR. Definicja osobotygodnia to jedna osoba pracująca przez cały tydzień na umowę o pracę przy pełnym etacie.

Pakiet roboczy	Wsparcie rozwoju FEC	Szacowany wysiłek
C1.1	Projekt i implementacja ogólnej klasy FESA z podklasami	~48 osobotygodnie
C1.2	Refaktoryzacja i modularyzacja wtyczki Eclipse do FESA	~24 osobotygodnie
C1.3	Prototypy do integracji automatyki przemysłowej i protokoły M2M	~12 osobotygodnie
C1.4	Wsparcie YOCTO	~16 osobotygodnie
C1.5	Wsparcie CI/CD	~16 Osobotygodnie
C2.1	Ulepszenia ogólnej stabilności i niezawodności	~24 osobotygodnie
C2.2	Aktualizacja orkiestracji i konfiguracji węzła	~48 osobotygodnie
C2.3	Aktualizacja administracyjnego graficznego interfejsu użytkownika	~12 osobotygodnie

3.3. Nakład pracy

Nakład pracy związany z opracowaniem odpowiednich pakietów roboczych wymienionych w tabeli w sekcji 3.2 został oszacowany przez dział systemu sterowania FAIR na około 196 osobotygodni. Oznacza to około 116 osobotygodni w dziedzinie kontroli sprzętu FEC i 80 osobotygodni dla systemu pomiarów i archiwizacji akceleratora.

Oszacowanie nakładu pracy zakłada, że dostawca posiada dużą wiedzę i doświadczenie w zakresie wykorzystywanych technologii i systemów. Oszacowany nakład nie obejmuje budżetu czasowego na zapoznanie się z wykorzystywanymi środowiskami programistycznymi i technologiami oprogramowania.

Wszystkie prace związane z pakietami roboczymi zostaną przydzielone i rozdzielone w razie potrzeby między systemy techniczne/pakiety robocze. Szczegóły zależą od bieżących postępów w rozwoju infrastruktury systemu sterowania akceleratorem i licznych interfejsów.

Zawartość każdej aktualizacji jest organizowana we współpracy z fizykami i ekspertami od konkretnych systemów w FAIR. Zakres, postępy i poniesiony nakład będą śledzone podczas regularnych spotkań. Programowanie i debugowanie musi odbywać się zdalnie, a częściowo na miejscu. Testy z mniejszymi systemami mogą być wykonywane na oddzielnym sprzęcie działającym w GSI w dedykowanym środowisku systemu kontroli integracji.

3.3.1. Obecność na kampusie FAIR/GSI

Większość działań związanych z rozwojem oprogramowania musi być wykonywana w ścisłej współpracy z ekspertami FAIR/GSI i inżynierami oprogramowania. W związku z tym wykonawca jest zobowiązany do zapewnienia określonej części czasu rozwoju w kampusie FAIR/GSI. *Dostęp do środowiska programistycznego, integracyjnego i produkcyjnego kontroli akceleratora jest ograniczony ze względu na bezpieczeństwo cybernetyczne i stabilność operacyjną. Pełny dostęp do oprogramowania jest dostępny wyłącznie w kampusie GSI/FAIR. Niezbędna jest zatem obecność na budowie co najmniej jednego odpowiedzialnego programisty. Dostęp zdalny do środowiska kontroli akceleratora poprzez ssh i tunelowanie portów usług jest dostępny w ramach zdalnego dostępu. VNC nie jest obsługiwane.*

Ten odsetek pracy do wykonania w FAIR/GSI w dużym stopniu zależy od konkretnego pakietu roboczego i zakłada się, że wynosi około 25% obecności, chyba że określono inaczej. Koszty te po stronie Wykonawcy winny zostać wliczone w kalkulację przedstawionej oferty.

3.3.2. Pomiary serwisowe i testowe na miejscu

Usługi obejmują obecność w placówce GSI, zwłaszcza w okresach wiązek, w których wykonywana jest większość programu testowego. Ciągła praca akceleratora jest zorganizowana w trzech 8-godzinnych zmianach. W przypadku typowego pomiaru testowego kilka zmian jest przypisanych w ciągu kilku dni lub tygodnia jako główny użytkownik.

Obecność na miejscu i harmonogram pracy akceleratora są przekazywane przez Dział Operacyjny FAIR/GSI. Chociaż roczny plan działania jest planowany z co najmniej 6-miesięcznym wyprzedzeniem, szczegółowe planowanie wsparcia na miejscu może mieć miejsce 10-14 dni przed pomiarem. *Wymaganie (zarezerwowanie) czasu pracy z wiązką akceleratora lub dedykowanego czasu testów akceleratora bez wiązki nie leży w zakresie odpowiedzialności wykonawcy. Od wykonawcy oczekuje się jednak dużego zaangażowania i elastyczności w przeprowadzaniu tych testów.*

3.3.3. Rozwój oprogramowania FEC

Akcelerator FAIR został zaprojektowany jako jeden z najbardziej złożonych i wszechstronnych obiektów badawczych jonów, protonów i antyprotonów na świecie. Dlatego też system sterowania akceleratorem jest również projektem oprogramowania na dużą skalę z systemem General Machine Timing w czasie rzeczywistym z synchronizacją czasu w całym kampusie FAIR/GSI w zakresie kilku ns. Wkład w ten system wymaga:

- Doświadczenia i dogłębnej znajomości systemów operacyjnych Linux, w tym rozszerzeń czasu rzeczywistego
- Wieloletniego doświadczenia w programowaniu w języku C++ oraz Python na potrzeby szybkiego prototypowania pakietów roboczych dotyczących oprogramowania do sterowania sprzętem FESA i powiązanego oprogramowania
- Znajomości standardowych narzędzi programistycznych, takich jak Eclipse i GIT
- [Znajomość protokołów MQTT, OPC UA](#)
- Doskonałej znajomości środowiska programistycznego Front-End Software Architecture (FESA) i doświadczenia w projektowaniu oprogramowania klasy FESA. [Zespół kontroli FAIR nie jest w stanie zapewnić kursów wprowadzających do środowiska FESA](#)
- Doskonała znajomość frameworka SILECS do integracji systemów opartych na PLC w środowisku systemu FESA

Całe powiązane oprogramowanie sterowania front-end musi być zaimplementowane przy użyciu standardowej struktury oprogramowania FAIR FESA (Front-End Software Architecture). Dalsze wymagania dotyczące systemu sterowania akceleratorem zostały podsumowane w następujących dokumentach.

Powiązane specyfikacje i wytyczne techniczne:

- 1174186 Standardowe specyfikacje systemu sterowania akceleratora [2] i odniesienia w nich:

- o Wytyczne dotyczące rozwoju
 - § 1235310 Wytyczne dotyczące rozwoju FESA [9]
 - § 1235719 Wytyczne dotyczące rozwoju graficznych interfejsów użytkownika GUI [10]
 - § 1235720 Wytyczne dotyczące rozwoju architektury oprogramowania [11]
 - § 1235721 Wytyczne dotyczące rozwoju w zakresie integracji sprzętu [12]
 - § 1240613 Wytyczne dotyczące rozwoju, Wytyczne dotyczące nazewnictwa systemów sterowania [13]
- o Wytyczne techniczne dotyczące łączności sieciowej Ethernet [4], sterowania urządzeniami [5], interfejsu sygnałów blokady i stanu [6], wymagań funkcjonalnych urządzeń [7] oraz zasad i przepisów projektowania elektrycznego [8]
- Szczegółowe specyfikacje:
 - o 1176025 Szczegółowe specyfikacje: framework oprogramowania FEC [14]
 - o 1176026 Szczegółowe specyfikacje: interfejs sprzętowy i sterowanie [15]
 - o 1176027 Szczegółowe specyfikacje: system zarządzania ustawieniami [16]
 - o 1176029 Szczegółowe specyfikacje: ogólny system określania czasu maszyny [17]
 - o 1176043 Szczegółowe specyfikacje klas urządzeń FEC [20]
 - o 1240229 Szczegółowe specyfikacje oprogramowania sterującego sprzętem [21]
 - o 1711614 Informacja techniczna o łączeniu systemów PLC z systemami front-end na bazie FESA [22]

3.3.3.1. Projekt i implementacja ogólnej klasy FESA z podklasami (WP C1.1)

Wykonawca ma zaprojektować i wdrożyć ogólną/standardową klasę FESA (oprogramowanie sterujące sprzętem) dla niezłożonych urządzeń ogólnego przeznaczenia, które mają ograniczoną liczbę stanów wejściowych i wyjściowych i są głównie sterowane przez zapisywanie wartości w rejestrach i monitorowane przez odczytywanie wartości z rejestrów.

Takimi urządzeniami są zawory próżniowe i pneumatyczne regulatory ciśnienia powietrza, które przenoszą sprzęt do wiązki. Co więcej niektóre klasy sprzętu do sterowania źródłem jonów mogą być objęte taką standardową klasą FESA, z możliwym rozszerzeniem o obsługę PLC (SILECS) w przyszłości.

Ogólna klasa FESA jest bazą kodu dla implementacji kilku podklas do sterowania określonego sprzętu. W ramach tego pakietu roboczego należy wdrożyć następujące podklasy:

- EZR (sterowanie RF dla źródła jonów EZR)
- HSAU (integracja specjalnych urządzeń iniektora HLI)
- ISAU (integracja specjalnych urządzeń źródła jonów)
- ISCP (pomiar wartości szczytowej prądu źródła jonów)
- ISDP (integracja nakładki źródła jonów)
- ISEE (kontrola elektrod ekstrakcyjnych źródła jonów)
- ISMO (napęd silnikowy źródła jonów)
- ISRM (monitor rentgenowski)
- ISSD (urządzenie Sputter Pulse Super)
- ISSP (Sputter Pulse)
- ISVT (integracja turbopompy molekularnej)
- MBUK (integracja wnęki sprężarki wiązki)

3.3.3.2. Refaktoryzacja i modularyzacja wtyczki Eclipse do FESA (WP C1.2)

Klasy FESA są tworzone jako pliki projektowe XML, z których generowana jest podstawowa struktura kodu C++. Plik projektu określa interfejs klasy, typy i struktury danych, informacje o czasie i wyzwalacze. Logika i komunikacja urządzenia są następnie implementowane w wygenerowanych plikach.

Wiele klas FESA można połączyć w jedną jednostkę wdrożeniową. Jednostka wdrożeniowa to opis XML, który jest używany do generowania kodu C++ dla klasy sprzętu, lokalizatora usług i konfiguracji wątków. Jednostka wdrażania określa warstwy współbieżności, wątki i sygnały zdarzeń. Jednostka

wdrożeniowa stanowi bazę do budowy pliku binarnego FESA. Plik binarny może być zbudowany jako "mieszany", zawierający zarówno część serwerową, jak i działającą w czasie rzeczywistym, lub oddzielnie.

Klasy FESA są tworzone przy użyciu opisu momentowego XML. Plik momentowy określa nazwę urządzenia, adresy sprzętowe i identyfikatory, mapowanie zdarzeń i kody czasowe.

Oba zadania, definicja projektu klas FESA i jednostka wdrożeniowa, są obsługiwane przez monolityczną wtyczkę Eclipse.

Wykonawca będzie odpowiedzialny za modularyzację, modernizację i utrzymanie wtyczki Eclipse.

Obecna wersja wtyczki jest zarówno dość przestarzała, jak i monolityczna w odniesieniu do pojedynczych kroków projektowania klasy FESA, tworzenia szablonów kodu i procesu wdrażania klasy FESA dla określonego urządzenia, w tym łączenia się z bazą danych FESA. Jest to główna wada wtyczki, ponieważ zmiany w platformie Eclipse, a także we frameworku FESA lub infrastrukturze usługowej związanej z FESA mają wpływ na całą wtyczkę, co sprawia, że poprawki są bardzo kłopotliwe. *Wtyczka Eclipse opiera się na podstawowym projekcie z komponentami specyficznymi dla laboratorium dla GSI i CERN. Zachowanie zgodności z rdzeniem w celu akceptowania przyszłych aktualizacji ogranicza możliwości refaktoryzacji. Celem jest przeniesienie odpowiedzialności za wdrożenia z Eclipse na narzędzia zewnętrzne i pozostawienie w Eclipse fazy projektowania i rozwoju. System kompilacji Yocto i przyszły system wdrażania mają na celu zmniejszenie ilości pracy wykonywanej przez wtyczkę, pozostawiając ją odpowiedzialną za konfigurację projektu, edycję projektu XML, walidację XML i edycję C++.*

Praca nad wtyczką polega głównie na utrzymaniu istniejącej funkcjonalności i zapewnieniu kompatybilności z przyszłymi wydaniem Eclipse. Jednym z celów refaktoryzacji jest system kompilacji i zależności, które wymagają przeróbki, aby ułatwić dostosowanie się do nowych wersji Eclipse. Nowa funkcjonalność będzie dotyczyć wspierania różnych celów kompilacji (Yocto/native) i przyszłych systemów wdrażania

Wykonawca przeprojektuje i/lub przekształci istniejącą wtyczkę w kilka modułów, zapewniając elastyczność dla przyszłych zmian i mając na uwadze, że nie wszyscy programiści piszący klasy FESA używają Eclipse do wszystkich etapów procesu rozwoju i wdrażania i mogą chcieć używać własnych skryptów (Python). Dlatego dobrym rozwiązaniem może być połączenie kodu Java dla części Eclipse z niektórymi modułami Pythona.

3.3.3.3. Prototypy do integracji automatyki przemysłowej i protokoły M2M (WP C1.3)

Zintegrowany system sterowania akceleratorem musi integrować i łączyć się z urządzeniami lub usługami dostarczonymi przez inne działy techniczne lub eksperymenty.

Oczekuje się, że niektóre z tych urządzeń lub systemów będą połączone za pomocą coraz bardziej popularnych standardowych protokołów Open Platform Communication/Unified Architecture (OPC/UA) lub lekkich protokołów sieciowych M2M (maszyna-maszyna). Wykonawca dostarczy koncepcję i prototyp urządzeń integrujących z systemem sterowania za pośrednictwem OPC/UA.

Dodatkowo ACO musi być w stanie komunikować się z urządzeniami za pośrednictwem lekkich protokołów komunikacyjnych wydawca-subskrybent, takich jak MQTT.

Przykłady:

- Integracja dozymetrów promieniowania
- Możliwe interfejsy do eksperymentów

Wykonawca dostarczy i zaprojektuje koncepcję i projekt prototypu do podłączenia urządzeń do systemu sterowania za pośrednictwem MQTT. *W momencie opracowywania interfejsy do urządzeń zostaną dostarczone przez GSI/FAIR lub zostaną określone wspólnie z dostawcą. całe niezbędne programowanie FPGA front-endu nie jest objęte zakresem tej umowy. Rozwój i niezbędne adaptacje zapewnia Dział Kontroli GSI.*

3.3.3.4. Wsparcie YOCTO (WP C1.4)

Wraz z nowym klastrem programistycznym ACC9, opartym na systemie operacyjnym Rocky 9 Linux [który jest stosowanym w FAIR](#), potrzebny jest YOCTO SDK i środowisko dla wbudowanego oprogramowania FESA działającego na płytach SCU używanych przez dział ACO.

Wykonawca zapewni wsparcie w zakresie dostarczania YOCTO jako celu w ramach procesu rozwoju i

wdrażania oprogramowania i klas FESA.

3.3.3.5. Wsparcie CI/CD (WP C1.5)

Wykonawca będzie wspierał automatyczną integrację i wdrażanie oprogramowania i klas FESA na wbudowanych kontrolerach, z integracją naszego repozytorium, zapewniając nocne kompilacje i testy regresji.

3.3.4. System archiwizacji

Celem systemu archiwizacji w GSI i FAIR jest gromadzenie, przechowywanie i wyszukiwanie danych generowanych przez poszczególne komponenty akceleratora lub usługi systemowe. Opracowany w 2019 roku stał się częścią stosu oprogramowania, który jest utrzymywany i obsługiwany przez zespół GSI/FAIR.

System archiwizacji jest w stanie wyszukiwać, filtrować, łączyć i wyświetlać dane szeregów czasowych otrzymane z różnych typów urządzeń. Możliwe jest również przechowywanie informacji z innych systemów, takich jak LSA, BTM itp. System opiera się na koncepcji urządzenia/właściwości/pól, które odpowiadają strukturze oprogramowania urządzenia (*FESA*, *DevAcc*) i oprogramowaniu pośredniczącemu (*CMW-RDA*, *DevAcc*) używanemu w GSI/FAIR.

System archiwizacji został zaprojektowany i zbudowany jako zbiór rozproszonych komponentów zsynchronizowanych za pośrednictwem usługi *zookeeper*. Obejmuje to administracyjny interfejs GUI do konfiguracji i konfiguracji systemu, a także przeglądarkę graficzną *DAVE* używaną do prezentacji danych użytkownikom. Do przechowywania rzeczywistych danych system wykorzystuje bazę danych *elasticsearch*.

Komponenty są zaimplementowane w języku programowania *JAVA*. Istnieją również pomniejsze komponenty *C++*. Komunikacja pomiędzy poszczególnymi komponentami realizowana jest za pomocą systemu pośrednictwa *Kafka*.

Wymagane są następujące prace rozwojowe:

- Szczegółowa analiza istniejącego oprogramowania systemu archiwizacji *MARS*, przegląd kodu (jeśli jest wymagany), debugowanie i definiowanie ulepszeń oprogramowania
- Refaktoryzacja kodu w celu wyeliminowania zależności od *zookeeper*
- Ogólne ulepszenia i aktualizacje usługi archiwizacji
- Poprawa ogólnej wydajności i odporności

Wkład w system archiwizacji wymaga następujących umiejętności technicznych:

- Wieloletnie doświadczenie i doskonała znajomość *Java 11/17* (*Open JDK*)
- Doświadczenie i wiedza w projektowaniu systemów rozproszonych
- Doświadczenie i znajomość *Grafany*
- Doświadczenie z następującymi technologiami i narzędziami: *Kafka*, *elasticsearch* (inne bazy danych *NoSQL* byłyby korzystne), kodowanie *Avro*, *JUnit*, *JSON*, *REST*, *Spring*, *JavaFX*
- Biegłość w korzystaniu z *Eclipse IDE*, *Maven*, *Git*

3.3.4.1. Ulepszenia ogólnej stabilności i niezawodności (WP C2.1)

Podczas korzystania z systemu archiwizacji w ciągu ostatnich kilku lat napotkano szereg przypadków użycia, w których system nie działał zgodnie z oczekiwaniami lub nie był w stanie poradzić sobie z występującymi problemami. Następujące punkty muszą zostać dodane przez współtwórców:

- Należy wdrożyć podstawowy mechanizm wykrywania utraty połączenia i ponownego uruchamiania połączeń. Utrata danych została zgłoszona przez użytkowników. Obecnie system polega na bazowym oprogramowaniu pośredniczącym w celu prawidłowego wykrywania utraty połączenia, co okazało się podejściem zawodnym.
- Wdrożenie rejestru schematów *AVRO*. System używa serializacji *AVRO* dla danych, wymaga znajomości schematu serializacji, który jest obecnie generowany z konfiguracji *zookeeper*.

Funkcja ta nie ułatwia bezpośredniego korzystania z Kafki przez inne grupy i wymaga dodatkowego komponentu do dostarczania danych użytkownikom.

- Przebudowa mechanizmu rejestrowania stanu. Obecnie system generuje znaczną ilość metryk i innych danych o stanie. Dane te są zapisywane w bazie danych. Nie ma jednak zaimplementowanego mechanizmu automatycznego usuwania tych danych, tj. wymagane są ręczne interwencje, aby nie przepelnić bazy danych. Istniejący mechanizm rejestrowania stanu musi zostać przebudowany z dodanymi opcjami kontrolowania ilości danych, a także usuwania danych po określonym czasie.
- Wdrożenie brakujących testów jednostkowych dla wszystkich komponentów. W niektórych komponentach wyraźnie brakuje testów jednostkowych. Aby jednostka wdrożeniowa była niezawodna, testy muszą być zaimplementowane dla wszystkich komponentów.

3.3.4.2. Ulepszenia orkiestracji i konfiguracji węzła (WP C2.2)

Obecnie system w dużym stopniu zależy od oprogramowania ZooKeeper, wykorzystując je do synchronizacji konfiguracji i koordynacji rozproszonych węzłów systemu archiwizacji. Ponieważ jednak nowsze wersje Kafki nie wymagają już usługi ZooKeeper, nie ma potrzeby utrzymywania zależności od niej dla oprogramowania archiwizującego. Aby usprawnić architekturę i infrastrukturę wymaganą do uruchomienia systemu archiwizacji, rozsądne może być odejście od ZooKeepera i zastąpienie go rozwiązaniami, które byłyby niezależne od usługi zewnętrznej. Decyzja w tej sprawie zostanie podjęta w trakcie realizacji zadania w momencie pojawienia się takiej potrzeby bez nie może w konsekwencji generować dodatkowych nakładów pracy po stronie Wykonawcy.

Ten pakiet roboczy obejmuje opracowanie rozproszonego magazynu konfiguracji i równoważenia obciążenia w celu dystrybucji obciążenia między aktywnymi węzłami oraz jego integrację z istniejącym systemem archiwizacji.

Wymagane są następujące prace rozwojowe:

- Zbadanie i przeanalizowanie bieżącego wykorzystania *ZooKeeper* w całym systemie
- Wybór i ocena możliwych alternatyw
- Wdrożenie rozwiązania i zintegrowanie go z istniejącym kodem systemu archiwizacji

3.3.4.3. Ulepszenia zarządzania (WP C2.3)

Graficzny interfejs administracyjny to interfejs oparty na JavaFX dostarczany wraz z kodem systemu archiwizacji w celu konfiguracji i monitorowania systemu archiwizacji. GUI jest obecnie najbardziej kompletnym narzędziem do konfiguracji i monitorowania systemu archiwizacji. Graficzny interfejs użytkownika zapewnia przegląd wdrożonych komponentów, komunikaty o błędach i ostrzeżenia, a także metryki danych, w tym rzeczywistą konfigurację urządzeń i właściwości do zebrania.

Istnieje kilka znanych problemów związanych z narzędziem administracyjnym, które muszą zostać rozwiązane w tym pakiecie.

- Brak wydajności, gdy skonfigurowanych jest wiele źródeł
- Brak opcji filtrowania określonych źródeł danych (np. tych ze zgłoszonymi awariami)
- Brak ostrzeżeń w określonych przypadkach użycia, np. konfigurowanie źródła, które już istnieje
- Prezentacja zdarzeń systemowych i powiadomień nie jest wiarygodna

3.4. Części zamienne

Części zamienne nie są częścią dostawy.

4. Specyfikacja systemu

Zestaw ogólnych wymagań ma zastosowanie do wszystkich rezultatów niniejszej specyfikacji:

R01) Obowiązują wszystkie zasady i przepisy podane w Ogólnej Specyfikacji FAIR [23].

- R02) Wszystkie zasady i przepisy podane w Standardowej Specyfikacji Systemu Sterowania Akceleratora [2]
- R03) Całe oprogramowanie do sterowników front-end i koncentratorów danych musi być napisane w języku FESA i musi być zintegrowane z ogólnym systemem sterowania FAIR, w tym np. strukturą LSA i głównymi usługami, takimi jak archiwizacja, analiza post-mortem itp.

4.1. Wymagania techniczne dotyczące komponentów systemu

4.1.1. Rozwój oprogramowania systemu sterowania

Pierwszym krokiem realizacji pakietu roboczego będzie zebranie, omówienie i uzgodnienie przez wykonawcę i FAIR/GSI wymagań technicznych dotyczących integracji oprogramowania w formie dokumentu projektu technicznego którego ostatecznym terminem będzie 15 miesięcy od momentu podpisania umowy. Wykonawca musi dostarczyć dokument projektu technicznego do zatwierdzenia przez Dział Sterowania przed rozpoczęciem wdrażania oprogramowania. W razie potrzeby należy dostosować klasy front-end FESA.

4.1.2. Środowisko rozwoju oprogramowania

Dział Sterowania GSI zapewni wykonawcy odpowiednie środowisko pracy (biurko z komputerem), dostęp do środowiska IT FAIR/GSI/sterowania akceleratora oraz środowisko programistyczne do odpowiednich struktur oprogramowania.

Należy pamiętać, że większość działań związanych z rozwojem oprogramowania musi być wykonywana w ścisłej współpracy z ekspertami FAIR/GSI, a większość wdrożeń i tak musi odbywać się na terenie kampusu FAIR/GSI.

W przypadku tworzenia poza siedzibą odpowiednie środowisko programistyczne musi być dostępne lub skonfigurowane przez dostawcę. Dostawca powinien jednak pamiętać i wziąć pod uwagę, że środowisko programistyczne dla ram systemu sterowania jest silnie powiązane z centralną infrastrukturą IT akceleratora z wieloma usługami, których nie można eksportować ani replikować.

W związku z tym, jeśli wykonawca nie posiada jeszcze aktualnego środowiska programistycznego FESA, należy rozważyć znaczny wysiłek w celu ustanowienia i przetestowania takiej instalacji poza kampusem FAIR/GSI. [Należy również pamiętać, że dział sterowania FAIR może zapewnić jedynie bardzo ograniczone wsparcie dla takiej działalności z powodu znaczącego obciążenia wynikającego z realizacji bieżących zadań z bardzo napiętym harmonogramem.](#)

4.2. Wymagania funkcjonalne (lista parametrów)

Wymagania funkcjonalne są definiowane osobno dla każdego pakietu roboczego.

5. Procedura

Proces realizacji komponentów FAIR, sprzętu lub oprogramowania, podzielony jest na dwie fazy: projektowania i produkcji. Każda faza kończy się bramką kontrolną jakości, tj. przeglądem lub testem akceptacyjnym.

Aby uzyskać ogólne informacje, czytelnik powinien zapoznać się ze specyfikacją ogólną [23], a szczegółowe informacje na temat przeglądów projektu koncepcyjnego i końcowego (CDR i FDR) można znaleźć w [24].

Szczegółowe informacje można znaleźć w dokumencie Standardowe Specyfikacje Systemu Sterowania Akceleratora (rozdział 4) [2].

5.1. Faza projektowania

Przegląd projektu koncepcyjnego (CDR): Na tym etapie należy ustalić koncepcję komponentu. Obejmuje to np. wszystkie wymagane symulacje, szacunki i obliczenia lub integrację komponentu z

jego środowiskiem pracy. Ponadto obejmuje przegląd wszystkich aspektów związanych z bezpieczeństwem.

Końcowy przegląd projektu (FDR): Końcowy przegląd określa gotowość produkcyjną komponentu. Na tym etapie muszą być dostępne wszystkie istotne informacje. Wszystkie dokumenty potrzebne do produktywnego użytkowania (programowanie, rysunki, uzgodnione plany kontroli itp.) są dostarczane przez dostawcę.

Szczegółowe informacje na temat przeglądów projektu koncepcyjnego i końcowego (CDR i FDR) można znaleźć w [24]. Ogólne wymagania muszą być dostosowane do potrzeb każdego indywidualnego typu projektu: oprogramowania, sprzętu mechanicznego lub elektroniki.

5.2. Faza produkcji

Aby uzyskać ogólne informacje, czytelnik powinien zapoznać się ze specyfikacją ogólną [23]. Wykonawca musi przygotować kompleksowy plan jakości (Q-Plan) oparty na normie ISO 10005 (norma przypisana do ISO9001) i przedłożyć go do zatwierdzenia.

6. Zapewnienie jakości, testy i akceptacja

Aby uzyskać ogólne informacje, czytelnik powinien zapoznać się ze specyfikacją ogólną [23]. Szczegółowe informacje można znaleźć w dokumencie Standardowe Specyfikacje Systemu Sterowania Akceleratora (rozdział 4) [2].

6.1. System zapewnienia jakości wykonawcy

Plan jakości producenta i dokumentacja testów akceptacyjnych podsumowują wszystkie informacje w celu zapewnienia wymaganej jakości komponentów [23].

Szczegółowe informacje na temat fabrycznych i lokalnych testów akceptacyjnych, FAT i SAT, można znaleźć w [25]. Ogólne wymagania muszą być dostosowane do potrzeb każdego indywidualnego typu projektu: oprogramowania, sprzętu mechanicznego lub elektroniki.

6.2. Fabryczny test akceptacyjny (FAT)

Szczegółowe informacje można znaleźć w dokumencie Standardowe Specyfikacje Systemu Sterowania Akceleratora (rozdział 4) [2]. Przed rozpoczęciem produkcji procedura FAT musi zostać uzgodniona na piśmie przez producenta i Dział Sterowania.

6.3. Test akceptacyjny na miejscu (SAT)

Szczegółowe informacje można znaleźć w dokumencie Standardowe Specyfikacje Systemu Sterowania Akceleratora (rozdział 4) [2]. Przed dostawą procedura SAT musi zostać uzgodniona na piśmie przez producenta i Dział Sterowania.

7. Dokumentacja

Aby uzyskać ogólne informacje, czytelnik powinien zapoznać się ze specyfikacją ogólną [23]. Szczegółowe informacje można znaleźć w dokumencie Standardowe Specyfikacje Systemu Sterowania Akceleratora (rozdział 4) [2]. Kompletna lista dokumentów musi zostać opracowana dla każdego komponentu i uzgodniona na piśmie przez producenta i Dział Sterowania.

8. Powiązane dokumenty stanowiące załącznik A2 do SWZ.

Dostępne w języku angielskim

- [1] 1174185 F-CS-BD-01e, „Standardowe specyfikacje diagnostyki wiązki”
- [2] 1174186 F-CS-C-01e, „Standardowa specyfikacja systemu sterowania akceleratora FAIR”
- [3] 1229368, F-TG-B-0.5e, "Identyfikacja komponentów i kody kreskowe"
- [4] F-TG-C-01e, "Łączność sieciowa Ethernet"

- [5] F-TG-C-02e, "Interfejsy sterowania urządzeniami"
- [6] F-TG-C-03e, "Interfejs sygnału blokady i stanu sprzętu"
- [7] F-TG-C-04e, "Wymagania funkcjonalne sprzętu"
- [8] F-TG-ET-01e, "Zasady i przepisy dotyczące projektowania elektrycznego"
- [9] 1235310, F-DG-C-01e "Wytyczne dotyczące rozwoju FESA"
- [10] 1235719, F-DG-C-02e "Wytyczne dotyczące rozwoju graficznych interfejsów użytkownika GUI"
- [11] 1235720, F-DG-C-03e "Wytyczne dotyczące rozwoju architektury oprogramowania"
- [12] 1235721, F-DG-C-04e "Wytyczne dotyczące rozwoju integracji sprzętu"
- [13] 1240613, F-DG-C-05e "Wytyczne dotyczące nazewnictwa systemów sterowania"
- [14] 1176025, F-DS-C-01e „Szczegółowe specyfikacje: framework oprogramowania FEC”
- [15] 1176026, F-DS-C-02e "Szczegółowe specyfikacje interfejsu i sterowania sprzętu"
- [16] 1176027, F-DS-C-03e „Szczegółowe specyfikacje: system zarządzania ustawieniami”
- [17] 1176029, F-DS-C-05e “Szczegółowe specyfikacje: ogólny system określania czasu maszyny”
- [18] 1176030, F-DS-C-06e "Szczegółowe specyfikacje odbiorników pomiaru czasu"
- [19] 1176039, F-DS-C-11e "System archiwizacji szczegółowych specyfikacji"
- [20] 1176043, F-DS-C-15e "Szczegółowe specyfikacje klas urządzeń FEC"
- [21] 1240229, F-DS-C-28e "Szczegółowe specyfikacje oprogramowania sterującego sprzętem"
- [22] 1711614, F-TN-C-013e „Informacja techniczna o łączeniu systemów PLC z systemami front-end na bazie FESA”
- [23] 1365092, F-GS-PMO-en, "Ogólna specyfikacja projektu akceleratora FAIR"
- [24] 1514206, Q-VA-QA-0006 "Przeglądy projektów (CDR i FDR)"
- [25] 1514174, F-VA-QUA-en-0025 "Testy akceptacyjne (FAT i SAT)"
- [26] 1176036, F-DS-C-08e "System blokad"
- [27] <http://www.cern.ch/white-rabbit> i <http://www.ohwr.org/projects/white-rabbit> White Rabbit Project