

Vorhaben

**@MOST-G**

Teilvorhaben

**Wireless Sensor Network Infrastructure and Software  
(SWAN-TGS)**

**&**

**SoA Air-Ground Communications Infrastructure  
(SO4A-TGS)**

# **Schlussbericht**



**TriaGnoSys GmbH**

---

## 0 Projekt- und Dokumentinformationen

### 0.1 Projektinformationen

<b>Verbundprojekt</b>	Aircraft Total Maintenance Operations, Solution & Technologies Program – Germany (@MOST-G)
<b>Verbundprojektleiter</b>	Airbus Operations GmbH
<b>Teilvorhaben</b>	Wireless Sensor Network Infrastructure and Software (SWAN-TGS) & SoA Air-Ground Communications Infrastructure (SO4A-TGS)
<b>Auftraggeber</b>	Bundesministerium für Wirtschaft und Technologie
<b>Projektträger</b>	DLR Projektträger Luftfahrtforschung und -technologie
<b>Förderkennzeichen</b>	20K0801F
<b>Projektlaufzeit</b>	01.12.2010 – 30.06.2012 (19 Monate)
<b>TriaGnoSys Projektleiter</b>	Dr. Markus Werner

### 0.2 Dokumentinformationen

Version	Revision	Datum	Änderung/Status	Editor
1	0	15.06.2013	Finale öffentliche Version	Markus Werner

<b>Autoren</b>	Nuria Riera, Emeric Dupont, Frank Kühndel, Alvaro Valcarce, Markus Werner	TriaGnoSys GmbH
<b>Geprüft</b>	Matthias Holzbock	TriaGnoSys GmbH
<b>Genehmigt</b>	Markus Werner	TriaGnoSys GmbH

## Inhaltsverzeichnis

<b>0</b>	<b>Projekt- und Dokumentinformationen</b>	<b>2</b>
0.1	Projektinformationen	2
0.2	Dokumentinformationen	2
	<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1</b>	<b>Kurzdarstellung des Vorhabens</b>	<b>4</b>
1.1	Aufgabenstellung	4
1.1.1	Aufgabenstellung des Verbundvorhabens	4
1.1.2	Aufgabenstellung des Teilvorhabens	5
1.2	Voraussetzungen der Durchführung	6
1.3	Planung und Ablauf	7
1.4	Stand der Technik	10
1.5	Zusammenarbeit mit anderen Stellen	12
<b>2</b>	<b>Eingehende Darstellung des Vorhabens</b>	<b>14</b>
2.1	Detaillierte Darstellung der Ergebnisse – Teilprojekt SWAN	14
2.1.1	AP 220: Interfaces	14
2.1.2	AP 310: Validation Platform Specification	14
2.1.3	AP 320: Lower Layer Node/Board Development	21
2.1.4	AP 330: Networking Software and Server Development	22
2.1.5	AP 340: Application/Client Software Development	24
2.1.6	AP 350: Localisation	25
2.1.7	AP 410: Test Cases	30
2.1.8	AP 420: Integration of Validation Platform	30
2.1.9	AP 440: Test&Lab Trials	34
2.2	Detaillierte Darstellung der Ergebnisse – Teilprojekt SO4A	36
2.2.1	AP 131: Funktionale Architektur	36
2.2.2	AP 132: Funktionale Architektur für Kommunikationsinfrastruktur	36
2.2.3	AP 141: Erstellung Funktionsmuster der Kommunikationsinfrastruktur	38
2.3	Wichtigste Positionen des zahlenmäßigen Nachweises	42
2.4	Notwendigkeit und Angemessenheit der geleisteten Arbeit	42
2.5	Voraussichtlicher Nutzen und Verwertung der Ergebnisse	43
2.6	Während des Vorhabens bekannt gewordener Fortschritt Dritter	45
2.7	Veröffentlichungen der Ergebnisse	45
	<b>Abkürzungsverzeichnis</b>	<b>46</b>
	<b>Literaturverzeichnis</b>	<b>47</b>

# 1 Kurzdarstellung des Vorhabens

Das vorliegende Dokument ist der Abschlussbericht zu zwei thematischen Teilprojekten, die TriaGnoSys nach der Umstellung des Verbundvorhabens „Aircraft Total Maintenance Operations, Solution & Technologies Program – Germany @MOST-G“ als eigenständiger Verbundpartner (und nicht mehr wie zuvor als Unterauftragnehmer von Airbus) fortführte und abschloss.

Die beiden Teilprojekte (SWAN-TGS und SO4A-TGS) waren formell und vertraglich in einem Teilvorhaben von TriaGnoSys gebündelt.

## 1.1 Aufgabenstellung

### 1.1.1 Aufgabenstellung des Verbundvorhabens

Das gemeinsame Ziel aller Forschungspartner innerhalb des Vorhabens @MOST-G ist das Erarbeiten neuer Lösungen für den Bereich der Flugzeugwartung und der technischen Unterstützung zur Aufrechterhaltung der Betriebstüchtigkeit einer modernen Flugzeugflotte. Ein Flottenbetrieb ist hierbei umso effizienter, je kürzer ein Flugzeug am Boden steht und je länger es in der Luft unterwegs sein kann. Ein Hauptgrund für unnötige Bodenstandzeiten sind überlange Wartungsarbeiten, die wiederum durch unflexible, manuelle Wartungsprozesse herbeigeführt werden können. Zwei verschiedene Aspekte können hierbei zur unnötigen Verlängerung von Wartungszyklen führen:

- unzureichende Datenlage über die durchzuführenden Wartungsarbeiten;
- ineffizientes Zusammenspiel der an der Wartungskette beteiligten Partner.

Das Teilprojekt **@MOST-SWAN** adressiert den Aspekt der unzureichenden Datenlage, die oftmals bereits vor Beginn der eigentlichen Wartungsarbeiten besteht. In einem konventionellen, voll manuellen Wartungsprozess geht im Schadensfall beispielsweise durch die nötige Problemdiagnose schon wertvolle Zeit verloren. Auf andere Art und Weise können Wartungskosten entstehen, wenn unnötige manuelle Inspektionen nach starren Wartungsintervallen durchgeführt werden, obwohl aktuell kein Wartungsbedarf entsteht. Hier setzt das im Teilprojekt @MOST-SWAN zu untersuchende drahtlose Sensornetzwerk an, welches für den zukünftigen Einsatz in Großraum-Flugzeugen gedacht ist. Die Datenlage über durchzuführende Wartungsarbeiten soll verbessert werden, indem durch die Knoten des Sensornetzwerks möglichst lückenlos Zustandsparameter einzelner Flugzeugkomponenten überwacht werden. Die stetige Auswertung von gemessenen Zustandsparametern vermindert den Wartungsaufwand, indem z.B. manuelle Diagnosearbeiten unterstützt und minimiert werden. Ein rechtzeitiges Erkennen unerwünschter Zustandsparameter von Flugzeugen kann zur Voraussage anstehender Wartungsoperationen eingesetzt werden und vermindert die Anzahl unnötiger und teurer Routinekontrollen.

Moderne Prozesse für die Flugzeugwartung erfordern eine ausgeklügelte logistische Kette, entlang derer die reibungslose Zusammenarbeit verschiedenster Partner ermöglicht sein muss. Luftfahrtgesellschaften, Flugzeug- und Systemhersteller, Wartungsdienstleister, Logistik-Anbieter, Flughafenbetreiber und viele andere sind auf den umfangreichen Austausch wartungsrelevanter Informationen angewiesen. Um eine effiziente Disposition von Wartungsressourcen zu gewährleisten, wird in der Architektur von Enterprise Software für

das Supply-Chain-Management das Prinzip einer dienstorientierten Architektur (engl. SOA = Service-Oriented Architecture) bereits eingesetzt. Das Teilprojekt **@MOST-SO4A** (Service-Oriented Architecture Application for Airlines & Aircraft) befasst sich daher unter Nutzung von SOA-Paradigmen – zusammen mit der Integration von ereignisbasierten Informationen – mit der Entwicklung von innovativen dienstorientierten Systemarchitekturen für den Flottenbetrieb, die zur Optimierung und Integration durchgehender Instandhaltungs- und logistischer Unterstützungsprozesse verwendet werden sollen. Diese sollen dann zudem für das gesamte Partnernetzwerk im Sinne einer effizienten, flexiblen und erweiterbaren Plattformlösung nutzbar gemacht werden. Als Basis hierfür kommt der Definition, Modellierung und Simulation dieser komplexen Prozesse und Dienste sowie der Erstellung einer entsprechenden integrierten Validationsplattform entscheidende Bedeutung zu.

### 1.1.2 Aufgabenstellung des Teilvorhabens

Die Aufgabenstellung und insbesondere die wissenschaftlichen-technischen Ziele der Teilvorhaben SWAN-TGS und SO4A-TGS sind eng mit denen des gesamten Verbundvorhabens verknüpft. Die spezifische Aufgabenstellung sowie die wichtigsten Ziel für die beiden Teilprojekte von TriaGnoSys sind im folgenden aufgeführt.

#### SWAN-TGS

Der größere Anteil der TGS-Arbeiten liegt im Teilprojekt SWAN. Insbesondere werden folgende wissenschaftlich-technischen Ziele verfolgt:

- Vervollständigung einer detaillierten Schnittstellenbeschreibung (mit Fokus Softwarearchitektur) für das gesamte Wireless Sensor Network (WSN);
- Erweiterung der funktionalen Softwarearchitektur hinsichtlich der neu hinzugekommenen location awareness Funktionalität;
- insbesondere im Zusammenhang mit der Lokalisierung eine Überprüfung/Konsolidierung einer synergetischen Zusammenführung mit SO4A;
- maßgebliche Beiträge zur Spezifikation des Technologiedemonstrators / der Validationsplattform;
- Design, Implementierung, Test und Validierung des Wireless Data Concentrator (WDC);
- Design, Implementierung, Test und Validierung der Kommunikationssoftware mit besonderem Fokus auf Routingprotokollen und Server-/Netzkontrollfunktionen;
- Entwicklung einer generischen Klientensoftware für den flexiblen adaptiven Einsatz in vielfältigen künftigen Sensor- und Aktuatoranwendungen/-szenarien;
- Aufbau und Betrieb des Referenz-Testlabors für das Gesamtsystem.

#### SO4A-TGS

Im Teilprojekt SO4A fokussiert sich TGS auf folgende wissenschaftlich-technischen Ziele:

- Entwurf der Infrastruktur. Dieser Entwurf umfasst die Definition von Basisdiensten, die für eine funktionierende Infrastruktur erforderlich sind. Dazu zählen u.a. ein einheitliches Dienstverzeichnis, Verfahren für die Nachrichtenübermittlung, sowie die Definition von Verfügbarkeit und Sicherheit;
- Forführung der Unterstützung bei Modellierung von air-ground links für ML Designer Implementierung;

- Beiträge zu Architektur, Spezifikation und Design mit Fokus air-ground Kommunikation:
  - Definition der funktionalen Architektur
  - Definition der Infrastruktur für SoA-Architektur
  - Dienstgüte-Definition
- Definition, Design und Implementierung eines erweiterten und eigenständigen SoA-Testbeds
  - mit Schnittstellen und Integrationsmöglichkeiten zu Simulationsumgebung und Unternehmensapplikationen
- Aufbau einer Lab-Umgebung für Szenario mit emulierten Flugzeugen, SoA-Repository, Service Broker, emulierte bzw. echte air-ground links, Bodenstationen, emulierte Nutzer/Beispielapplikationen; Fokus auf generischen SoA-Mechanismen

Demonstration und Optimierung der SoA-basierten Kommunikation zwischen Flugzeug und Boden

## 1.2 Voraussetzungen der Durchführung

TriaGnoSys hat sich seit der Gründung der Firma 2002 als spezialisierter Anbieter vor allem von höchst innovativen aeronautischen Kabinenkommunikationslösungen auf dem Weltmarkt etabliert. Dabei liegt der Fokus bisher auf aeronautisch zertifizierter Kommunikationssoftware und bei Diensten vornehmlich für Passagiere, zunächst in Großraumflugzeugen, zunehmend aber auch im Geschäftsflugsegment. Die Anbindung lokaler Funkzellen an Bord (GSM, 3G, WLAN, etc.) erfolgt dabei vor allem über Satellitenverbindungen, je nach Szenario aber auch über direkte Flugzeug-Boden-Verbindungen (*direct-air-to-ground links*).

In laufenden Projekten dehnt TriaGnoSys die Geschäftstätigkeit zunehmend Richtung Systemintegration mit verstärkter Hardwareeinbindung aus; außerdem werden die zur Passagierkommunikation komplementären Anwendungsbereiche Cockpit-Kommunikation/Flugverkehrsführung, *In-Flight-Entertainment* (IFE) sowie diverse Datenkommunikation (u.a. Wettersensordaten) adressiert. Bei diesen Entwicklungen zur effizienten Integration unterschiedlichster Anwendungen ist die von TriaGnoSys für Airbus maßgeblich entwickelte Instanz des *Communication Manager* eine hervorragende Plattform. Außerdem bilden hocheffiziente Kompressions- und Multiplexinglösungen wie das TriaGnoSys-Produkt *VoCeM* eine sehr gute Basis für weitergehende Dienstintegration.

Mit Blick auf die in SWAN genutzten Sensornetze bringt TriaGnoSys eine langjährige Expertise und Kernkompetenz in modernen Netzprotokollen und –technologien ein. Insbesondere bei Routingverfahren in Netzen mit dynamischer Topologie kann TriaGnoSys auf weitreichende eigene Forschungsarbeiten (Satellitennetze, Ad-Hoc-Netze) und innovative Lösungen aufbauen. Anwendungsorientierte Forschung im Bereich Simulation und Optimierung von Netzprotokollen und Algorithmen ist bereits in vielen und vielfältigen Forschungsprojekten eine Kernrollen von TriaGnoSys gewesen.

Im Bereich Service-Oriented Architecture (SoA) ergeben sich für TriaGnoSys weitreichende Synergien mit vergangenen und laufenden Projekten und Produktentwicklungen in den Bereichen (i) neuartige Anwendungen für zukünftige Flugverkehrskontrolle (SoA-basierte SWIM-Anwendungen – System Wide Information Management), (ii) dienstorientierte Wetter- und Meteo-Applikationen für Flugzeuge mit/ohne Integration von Wettersensoren und (iii) verteilte intelligente und nachhaltigkeitsorientierte Logistik- und Produktionsprozesse.

In beiden Teilprojekten, SO4A und SWAN, nimmt TriaGnoSys eine führenden Rolle bei Design, Entwicklung, Test&Verifikation und Integration der Software ein. Softwareentwicklung nach aeronautischen Prozessen ist bei TriaGnoSys langjähriger Standard, und die Erfahrungen aus der Entwicklung mehrerer erfolgreicher Softwareprodukte fließt typischerweise auch in Forschungsprojekte wie @MOST mit vorkommerziellen und/oder prototypischen Implementierungen ein.

### 1.3 Planung und Ablauf

Die Aufteilung der Arbeitspakete und Beschreibung der Verantwortlichkeiten ist das Ergebnis eines gründlichen Abstimmungsprozesses zwischen allen beteiligten Partnern innerhalb von @MOST-SO4A bzw. @MOST-SWAN – sowohl in der ursprünglichen Projektausarbeitung wie auch der Projektumstellung – und stimmt deshalb mit den Angaben in der Vorhabensbeschreibung des Verbundführers Airbus überein. Die detaillierten Beschreibungen der Arbeitspakete enthalten

- Ziele des Arbeitspakets,
- Aufgabenbeschreibung,
- Verantwortlichkeiten und Leistungen der Partner.

Abbildung 1 und Abbildung 2 zeigen die inhaltliche Strukturierung des Verbundvorhabens mit seinen beiden Teilprojekten SWAN und SO4A, Abbildung 3 und Abbildung 4 die zugehörige Planung des zeitlichen Ablaufs inklusive Meilensteinen. Aufgrund der vertraglichen Projektumstellung in 2010 verzögerte sich das Projekt im weiteren Verlauf um einige Monate.

Wie aus den Arbeitspaketstrukturen hervorgeht, waren die verschiedenen Arbeiten der Verbundpartner im Gesamtvorhaben teilweise sehr eng aufeinander abgestimmt und komplementär. Um eine möglichst reibungslose und effiziente Projektarbeit zu gewährleisten, war TriaGnoSys mit seinen Teilvorhaben deshalb in der großen Mehrheit der Arbeitspakete und Unter-Arbeitspakete beteiligt.

Die Schwerpunkte der Arbeiten von TriaGnoSys innerhalb des Verbundes lagen dabei in folgenden Arbeitspaketen (fett diejenigen unter TriaGnoSys-Führung):

SWAN-TGS APs: 220, 310, 320, **330**, 340, 350, **420**, 430, **440**;

SO4A-TGS APs: (130), 131, **132**, (140), **141**.

Die inhaltliche Aufgabenstellung der jeweiligen Arbeitspaketanteile für TriaGnoSys ist für die bessere Lesbarkeit jeweils im Zusammenhang mit dem Bericht über die erzielten Ergebnisse in den entsprechenden Unterabschnitten der Abschnitte 2.1 (SWAN) und 2.2 (SO4A) aufgeführt.

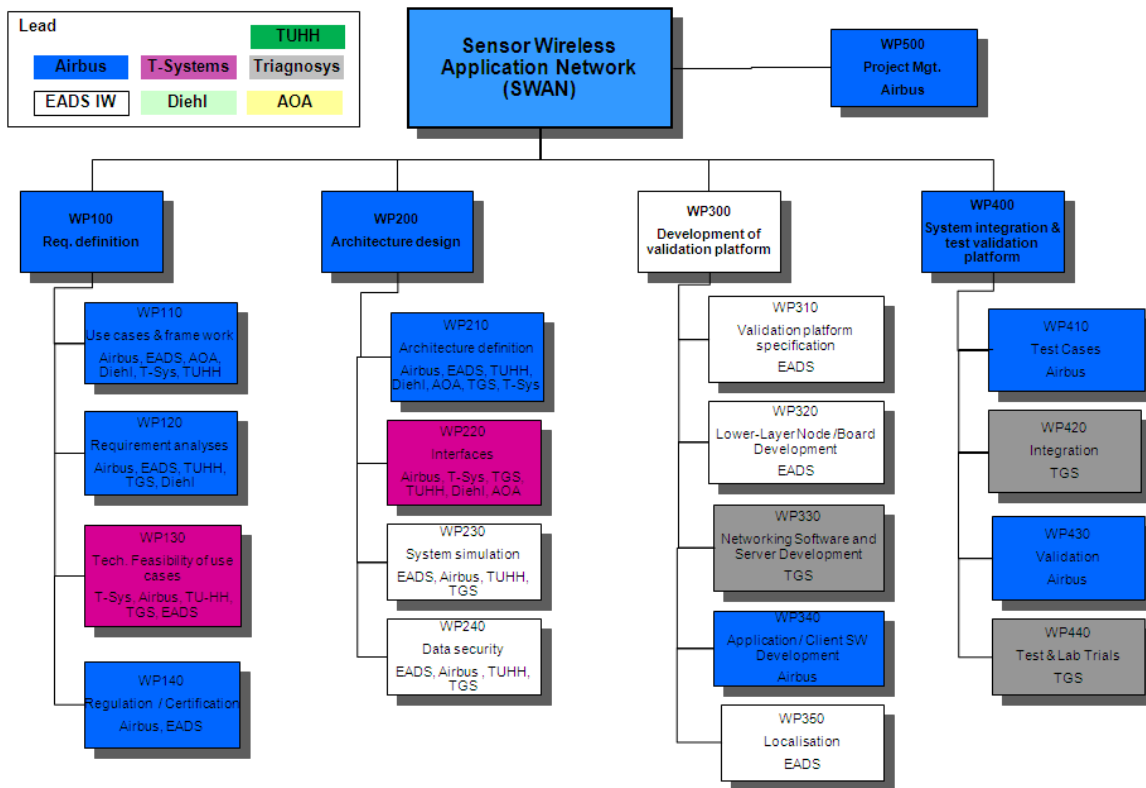


Abbildung 1: Arbeitspaketstruktur des Teilprojekts SWAN

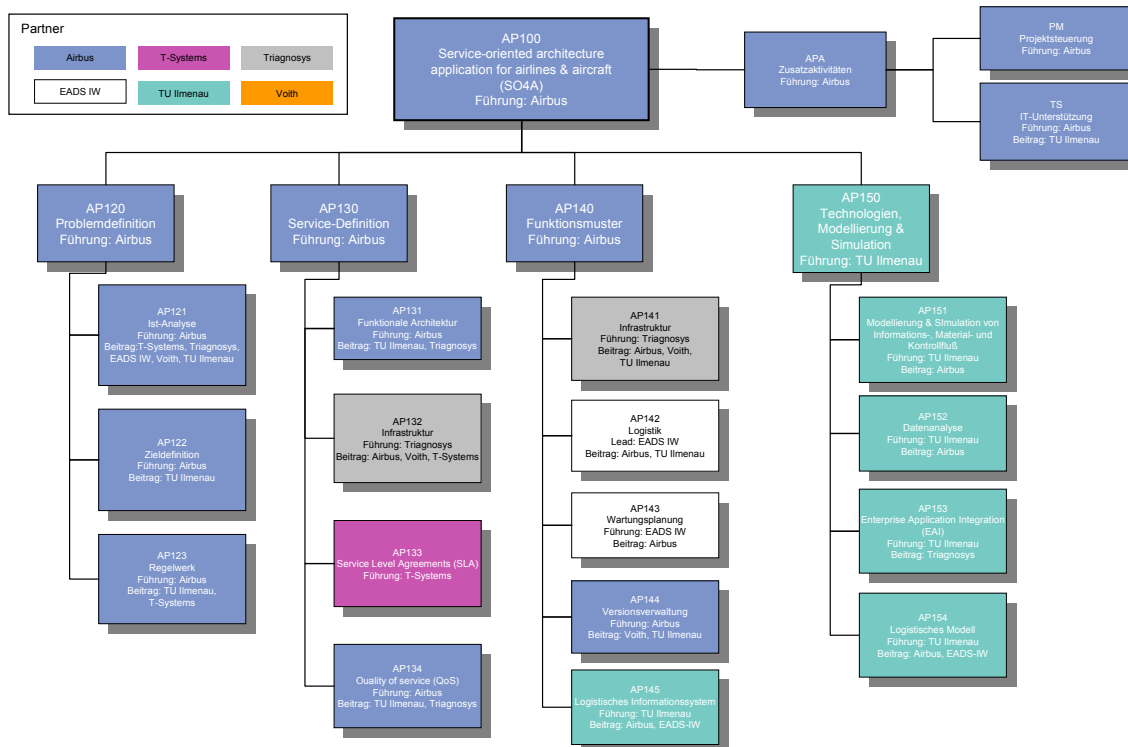


Abbildung 2: Arbeitspaketstruktur des Teilprojekts SO4A



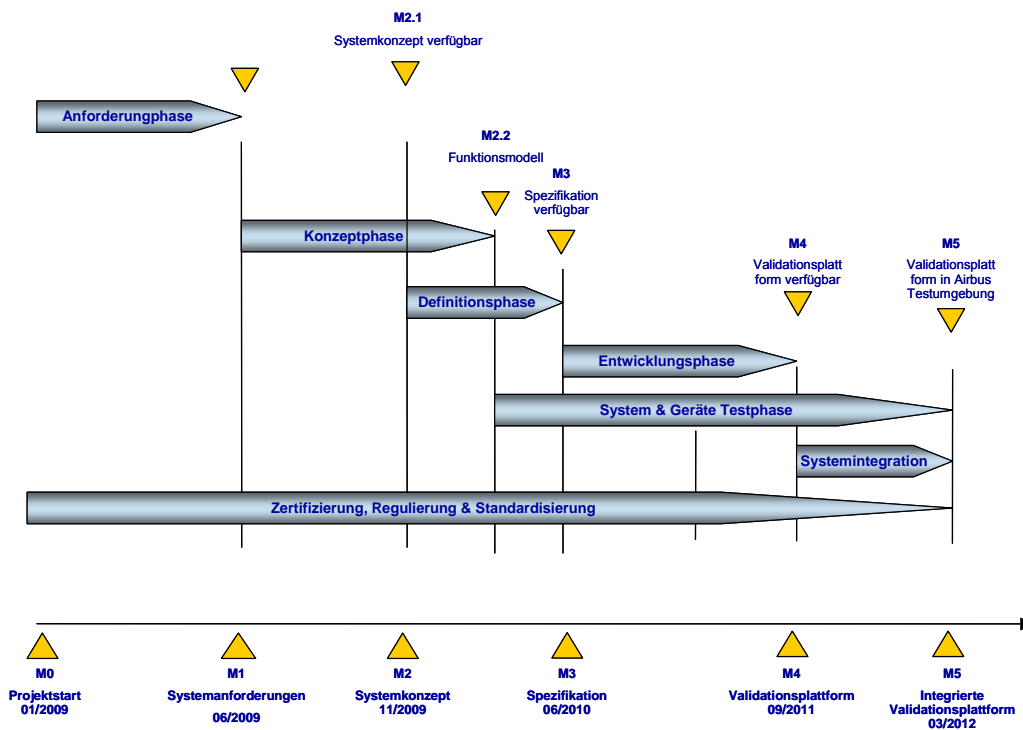


Abbildung 3: Zeitablauf-Diagramm und Meilensteine des Teilprojekts SWAN

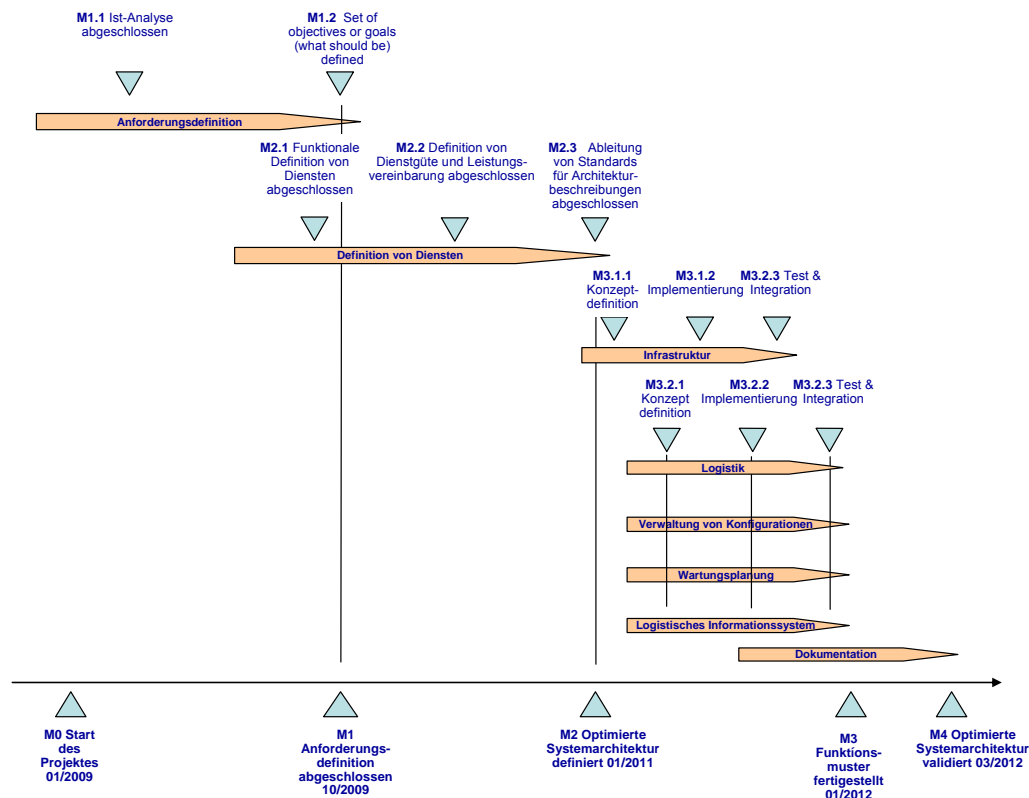


Abbildung 4: Zeitablauf-Diagramm und Meilensteine des Teilprojekts SO4A

## 1.4 Stand der Technik

Der folgende Überblick über den Stand der Wissenschaft und Technik bezieht sich im wesentlichen auf das Verbundvorhaben, in welches das Teilvorhaben eng integriert ist; entsprechend ist dieser umfassende Stand der Technik die Ausgangsbasis insbesondere auch des Teilvorhabens.

### @MOST-SWAN

Sensor-Netze bestehen aus stationären und mobilen Sensorknoten, in denen eine geeignete Sensormesstechnik, eine Datenverarbeitungseinheit, eine Energieversorgung und drahtlose Funkschnittstelle auf kleinstem Raum integriert sind. Diese Sensorknoten können beispielsweise an unzugänglichen Stellen angebracht sein und vollkommen autark über längere Zeiträume Umgebungswerte messen oder physikalische Prozesse überwachen [1]. Mit Sensor-Netzen lassen sich Anwendungen realisieren, die durch Miniaturisierung und Einbettung von Mikroelektronik in andere Objekte sowie ihre Vernetzung und Allgegenwart im Alltag gekennzeichnet sind.

Sensor-Netze unterscheiden sich von traditionellen Rechnernetzen durch Einschränkungen der Ressourcen wie beispielsweise Speicherkapazität, Übertragungsleistung und Energieversorgung. Viele Anwendungen erlauben keinen Wechsel der Batterien. Deshalb ist eine Unabhängigkeit des Sensor-Netzes gefordert. Dies kann beispielsweise durch den Einsatz von regenerativen Energien in den Sensorknoten erreicht werden (*energy harvesting*). Das Einsatzgebiet regenerativer Energiequellen ist zur Zeit noch stark eingeschränkt, teils weil entsprechende Gewinnungsverfahren noch nicht existieren oder weil die produzierten Energiemengen zu gering sind oder nicht kontinuierlich sondern nur sporadisch gewonnen werden können. Einen Architekturvorschlag für einen Sensorknoten auf Basis von energy harvesting findet man in [2]. Aus diesen Gründen beziehen Sensorknoten in der Regel ihre Energie aus Batterien, die beschränkte Größe erlaubt aber nur sehr kleine Batterien mit geringer Kapazität. Der langlebige Betrieb solcher Sensorknoten erfordert deshalb einen extrem ökonomischen Umgang mit der Ressource „Energie“.

Die größten Energieverbraucher in den betrachteten Sensorknoten sind zum einen die Sensoren selber, die einen physikalischen Effekt vermessen sollen und zum anderen die drahtlose Kommunikation. Der Energieverbrauch durch die Sensorinterface-Elektronik oder den Prozessor fällt dagegen kaum ins Gewicht. Dabei ist zu beachten, dass in vielen Fällen die Bereitschaft des Funkmoduls zum Empfang von Nachrichten mehr Energie benötigt als das eigentliche Senden von Daten. Aus diesem Grund wurden zahlreiche Verfahren zur energieeffizienten Medienzugriffskontrolle entwickelt (MAC-Protokolle). Vertreter dieser Art von MAC-Protokollen sind B-MAC, WiseMAC, LMAC und TRAMA. Der aktuelle Forschungsstand auf diesem Gebiet ist in [3] zusammengefasst. MAC-Protokolle reduzieren den Energieverbrauch durch die Einführung von Schlaf-Wach-Zyklen auf Kosten des Durchsatzes und der Latenzzeit. In den Schlafphasen werden alle aktiven Komponenten im Sensorknoten einschließlich des Funkmoduls abgeschaltet.

Sollen Messwerte von verschiedenen Sensorknoten zeitlich miteinander kombiniert werden, so ist eine Synchronisation der Uhren notwendig. Die geforderte Genauigkeit hängt von der konkreten Anwendung ab und kann in der Größenordnung von Millisekunden bis hin zu einigen Mikrosekunden liegen. Derartige Genauigkeiten sind mit speziellen Algorithmen zu erreichen, wobei der Aufwand für die Zeitsynchronisation und damit auch der Energiebedarf mit der Netzgröße steigt. Mit aktuellen Verfahren kann eine Genauigkeit zwischen zwei Knoten in der Größenordnung von weniger als 100  $\mu$ s erreicht [4] werden.

Zur Unterstützung der Erstellung von Anwendungen von Sensor-Netzen sind in den letzten Jahren auf mehreren Ebenen Fortschritte erzielt worden. Mit „TinyOS“ steht ein quelloffenes

Betriebssystem für Sensor-Netze zur Verfügung, welches die spezifischen Belange, die sich aus der Ressourcenbeschränkung ergeben, berücksichtigt [5]. Die Interpretation der in einem Sensornetz verteilt gemessenen Daten ist Bestandteil vieler Anwendungen. Hierzu gibt es erste Ansätze, die auf Datenbank-Sichtweisen basieren. So interpretiert der TinyDB-Ansatz das Sensornetz als eine verteilte Datenbank, die durch Sensoren dynamisch mit Daten der realen Welt gefüllt wird [6]. Mittels einer SQL-ähnlichen Sprache kann man Anfragen an das Sensornetz stellen.

Wie in drahtgebundenen Netzen erfordert die Kommunikation der Knoten untereinander auch in Sensor-Netzen Verfahren zum Routing von Nachrichten. Die für drahtgebundene Netze entwickelten Verfahren lassen sich nicht einfach übertragen, da dort die Frage des Energieverbrauchs nicht berücksichtigt wird. Von besonderer Bedeutung für Sensor-Netze sind geografische Routing-Verfahren. In vielen Anwendungsszenarien interessiert sich der Benutzer gezielt für Messdaten einer bestimmten geografischen Region. Das Netzprotokoll muss Anwendungen dabei die Aufgabe abnehmen, die betroffenen Knoten ausfindig zu machen und Nachrichten an sie weiterzuleiten. Einen Überblick über den aktuellen Stand der Routing-Verfahren in Sensor-Netzen findet man in [7]. Neuere geografische Routing-Algorithmen sind in [8] beschrieben.

Sensor-Netze müssen ihre Aufgabe in robuster Weise erfüllen. In lebenswichtigen Anwendungen ist darüber hinaus Verlässlichkeit und Fehlertoleranz gefordert, d.h. ein Sensornetz sollte möglichst auch bei Ausfall einzelner Komponenten oder beim Auftreten zu erwartender Fehler wie Störungen des Funkkanals noch Funktionsgarantien erfüllen. Einen Ansatz zur Realisierung von Fehlertoleranz in drahtlosen Sensor-Netzen wird in [9] beschrieben.

Aufgrund der anvisierten Einsatzgebiete müssen Sensor-Netze ohne Infrastruktur auskommen, Sensorknoten müssen sich also spontan vernetzen. Techniken aus dem gegenwärtig intensiv erforschten Bereich der Ad-hoc-Netze und der Selbstorganisation spielen daher eine bedeutende Rolle. Ebenso ist Skalierbarkeit eine wichtige Anforderung, da Sensor-Netze Tausende oder gar Millionen von Knoten umfassen können. Die Größe der Netze und die Unzugänglichkeit der Knoten selbst verhindert, dass einzelne Knoten gezielt manuell konfiguriert oder administriert werden können, Sensor-Netze haben daher selbstkonfigurierend und selbstadministrierend zu sein. Einen Überblick über den aktuellen Forschungsstand findet man in [10].

Für viele Anwendungen von drahtlosen Sensor-Netzen ist Sicherheit ein wichtiges Kriterium. Die mannigfaltigen Sicherheitsrisiken, denen solche Netze ausgesetzt sind, haben ihre Wurzeln in der drahtlosen Kommunikation. Die reduzierten Rechenleistungen von Sensorknoten sind der wesentliche Grund, weshalb konventionelle Sicherheitslösungen in diesem Umfeld nur sehr eingeschränkt anwendbar sind. Den aktuellen Stand der Forschung beschreiben Liu und Ning [11].

Eine der neueren Aktivitäten der Arbeitsgruppe für Wireless Personal Area Networks (WPANs) im Standardisierungsvorhaben IEEE 802.15.4 beschäftigt sich mit einer neuartigen Funkchip-basierten Technologie für die genaue Ortung der Position von drahtlosen Sensorknoten [12]. Im Gegensatz zu bereits kommerziell erhältlichen Lokalisierungstechnologien, die auf die Analyse der Signalstärke oder der Laufzeit eines Funksignals setzen, basiert der in der Arbeitsgruppe präsentierte Hardware-Prototyp der Firma Atmel auf der Analyse der Phase des von einem Knoten zum nächsten und wieder zurück gesendeten Funksignals. Die in [12] vorgebrachten Vorteile der Technologie versprechen eine schnellere und energie-effizientere Sensorortung. Durch eine Mittelung der verwendeten Signale über mehrere Frequenzen sollen aber vor allem auch negative Einflüsse wie die vor allem in typischen Innenräumen unvermeidliche Multipath-Ausbreitung vom Funkchip teilweise kompensiert werden können. Die damit möglicherweise erreichbare hohe Ortungsgenauigkeit verspricht erstmalig die Realisierbarkeit ganz neuer ortssensitiver

Anwendungsklassen im eng begrenzten und mit vielen Funkhindernissen versehenen Flugzeuginnenraum. Seit kurzem hat der Forschungsverbund erstmalig Zugriff auf einen Prototypen dieser neuen Technologie. Die in @MOST-SWAN versammelten Partner haben sich darum entschlossen, das Potential der Technologie für zukünftige ortungsabhängige Wartungsprozesse gemeinschaftlich zu erforschen.

### **@MOST-SO4A**

Der Entwurf komplexer Systeme stellt eine Herausforderung für die Entwicklung zukünftiger Systeme dar. Aufgrund der stetig zunehmenden Komplexität ist es heute i.d.R. nicht mehr möglich, den Systementwurf durch einen funktionalen Aufbruch des Systems und dem anschließenden Detailentwurf von einzelnen Systemkomponenten durchzuführen, da sich normalerweise Effekte, die durch die Kopplung der einzelnen Systemkomponenten im Sinne des Gesamtsystems entstehen, erst während der Test- und Integrationsphase zeigen. Eine Änderung des Entwurfs in dieser Phase ist jedoch üblicherweise mit hohem Aufwand und Kosten, sowie erheblichen Zeitverzögerungen verbunden.

Man ist daher bestrebt, diese Effekte bereits während der Entwurfsphase zu studieren, um bereits frühzeitig zu einer optimierten Systemarchitektur zu kommen. Dazu werden i.d.R. Modelle erstellt, die mit Hilfe eines Simulators ausführbar gemacht werden. Bereits im Verlauf des Systementwurfes können so Kopplungseffekte und verschiedene Architekturvarianten untersucht werden. Die Modelle müssen dabei drei wesentliche Aspekte eines Systems beschreiben:

- Informationsfluss,
- Kontrollfluss,
- Produktfluss.

Nur wenn alle diese Aspekte abgebildet werden, ist es möglich, ein integriertes Modell zu erstellen, anhand dessen die Architektur des zu entwerfenden Systems optimiert werden kann.

Für das Teilvorhaben von TriaGnoSys ist insbesondere der Informationsfluss im Rahmen der Kommunikationsinfrastruktur des Gesamtsystems bedeutsam. In den letzten Jahren ist in verschiedensten Programmen wie SESAR sowie diversen EU- und nationalen Projekten ein deutlicher Trend hin zu SOA-Architekturen auch im Bereich der aeronautischen Kommunikation zu beobachten. Dabei nimmt insbesondere das Konzept des sogenannten *System Wide Information Management (SWIM)* eine zentrale Rolle ein. Darin werden u.a. SOA-Ansätze konzeptionell beschrieben, wie sie im Bereich der aeronautischen Kommunikation für Flugverkehrsführung sowie für administrative und operationelle Aufgaben von Fluglinien in Zukunft eingesetzt werden sollen. Diese Grundlagen sind weitgehend konzeptioneller Natur, so dass das Teilprojekt SO4A mit der Ambition eines Software-Testbeds für ausgewählte Kommunikationsaufgaben hinsichtlich des detaillierten Designs sowie der Implementierung ziemliches Neuland betritt.

## **1.5 Zusammenarbeit mit anderen Stellen**

Im Verbundvorhaben waren die verschiedenen Arbeiten der Verbundpartner zum großen Teil sehr eng aufeinander abgestimmt und komplementär. Insofern gehen viele Einzelheiten der Arbeitsteilung und der Zusammenarbeit der Verbundpartner aus der Darstellung der Projektplanung sowie der Arbeitspakete in Abschnitt 1.3 hervor.

Die wichtigsten Aspekte der Arbeitsteilung im Forschungsverbund und die Zusammenarbeit mit den jeweiligen Partnern lassen sich aus Sicht von TriaGnoSys wie folgt zusammenfassen.

## @MOST-SWAN

- In AP320 erfolgt die WDC-Entwicklung verantwortlich durch TGS (Task322), wobei eine enge Abstimmung mit Partner EADS-IW an der Schnittstelle zu den Sensorknoten (Task321) erfolgt.
- TriaGnoSys übernimmt die Führung und weitgehende Implementierungsarbeiten in AP 330 „Netzwerksoftware und Serverentwicklung“, wobei folgende spezifische Zuarbeit von Partnern vorgesehen ist:
  - TUHH: Einbringung der Energiemanagementfunktionalität, Unterstützung der Implementierung;
  - EADS-IW, Diehl, AOA: Unterstützung bei der Implementierung der Schnittstelle „Klientensoftware“ – „Systemsoftware“
  - EADS-IW: Unterstützung bei der Implementierung der Schnittstelle „Kontinuierliche Erfassung eines mobilen Gerätes“.
- In AP340 entwickelt TGS parallel, aber abgestimmt auf die spezifischen Anwendungen der Partner AOA, Diehl und EADS-IW, eine generische Klientensoftware.
- In AP350 stellt TGS sicher, dass die unter Führung von EADS-IW und mit Airbus-Algorithmen ergänzte Lokalisierungsfunktion und –technologie auch Server-seitig unterstützt und implementiert wird; es ergibt sich eine klare web-basierte Softwareschnittstelle insbesondere zum Maintenance-PDA-Klienten von EADS-IW.
- TGS führt die Integration aller Soft- und Hardwareanteile zu einer Validationsplattform in AP420 und arbeitet dabei in mehreren klar definierten Phasen mit allen Verbundpartnern zusammen.
- TGS führt in AP440 verantwortlich alle begleitenden Geräte- und Systemtests sowie Trials in der Integrationsumgebung durch, die sich wiederum eng an die Arbeiten in AP420 anlehnen und damit auf der kontinuierlichen Abstimmung mit den jeweiligen Partnern beruhen.

## @MOST-SO4A

- In AP132 führt TGS die Definition der Kommunikationsinfrastruktur für eine Service-orientierte Architektur und wird dabei eng von Airbus und TU Ilmenau unterstützt.
- Die Implementierung der in AP132 definierten Subsysteme erfolgt unter Führung von TGS in AP141 mit Unterstützung/Zuarbeit von Airbus und TU Ilmenau:
  - Implementierung einer Dienste-Infrastruktur zur Diensteintegration
  - Demonstration der Dienstintegration und deren Strategien
- Implementierung einer erweiterten und eigenständigen SOA-Testumgebung im Labor; diese weist klare Schnittstellen zu ML-Designer-Modellen und der Simulationsumgebung von TU Ilmenau und Airbus auf; insbesondere werden Anforderungsdaten aus dem Flottenmodell der TU Ilmenau zur Ansteuerung des Kommunikations-Testbeds verwendet. Im Gegenzug werden vor allem relevante Parameter wie die Verzögerungszeiten von Nachrichten, die aus dem Testbed gewonnen wurden, in die übergeordnete ML-Designer-Modellierung eingespeist. Damit wird das Modell realistischer, und der Einfluss der Kommunikation auf dem Gesamtprozess kann simuliert werden.

Über die gesamte Projektlaufzeit war das Verbundvorhaben von einer engen und zielgerichteten Zusammenarbeit aller Partner gekennzeichnet.

## 2 Eingehende Darstellung des Vorhabens

### 2.1 Detaillierte Darstellung der Ergebnisse – Teilprojekt SWAN

#### 2.1.1 AP 220: Interfaces

Das Arbeitspaket wurde mit Zuarbeit von TriaGnoSys als Unterauftragnehmer von Airbus bereits vor dem formellen Vertragsbeginn des hier umgestellten Vorhabens abgeschlossen.

#### 2.1.2 AP 310: Validation Platform Specification

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

Mit Fokus auf WDC und Server trägt TriaGnoSys zur Spezifikation und Auswahl der Hardwarekomponenten für die Validierungsplattform bei. Die Software-Spezifikation für die Validierungsplattform wird im Verbund mit allen anderen Partnern von TGS geführt.]

Die **Hardware-Spezifikation** für die SWAN-Validierungsplattform wurde in mehreren Iterationen mit den Partnern erstellt. Dabei legte TriaGnoSys insbesondere die Hardware-Komponenten für WDC (Koordinator-Knoten, *Wireless Data Concentrator*) und Server fest:

- Der WDC ist ein *BRIK System Fanless Embedded Server Computer*, der einen Koordinator-Knoten mit einer oder zwei Antennen beherbergen kann. Dieser Koordinator-Knoten ist über eine USB-Schnittstelle anzusprechen.
- Der Server ist ein *FujitsuSiemens Laptop E8020*.

Details und Bilder von den entwickelten WDCs sind weiter unten bei AP320 zu finden.

Die **Software-Spezifikation** für die SWAN-Validierungsplattform wurde unter TGS-Führung in Zusammenarbeit mit den Partnern in mehreren Iterationen erstellt und in der konsolidierten Version des Deliverable D310.2 (Software-Spezifikation) fixiert.

Das wesentliche Resultat dieses umfangreichen Prozesses ist auf der höchsten Ebene in der graphischen Darstellung der SWAN-Software-Architektur in Abbildung 5 dargestellt. Es zeigt die wesentlichen Blöcke

- SWANControlManagement
- HMI Server
- ClientApplication
- Phase-based/RSSI-based Localisation
- WDCControlManagement
- SensorNode
- Coordinator Node

mit ihren jeweiligen Modulen, Interfaces und Detailinfos zur Funktionalität.

Die Zuordnung der verschiedenen Software-Komponenten/Applikationen zu Hardware-Komponenten ist Abbildung 6 zu entnehmen.



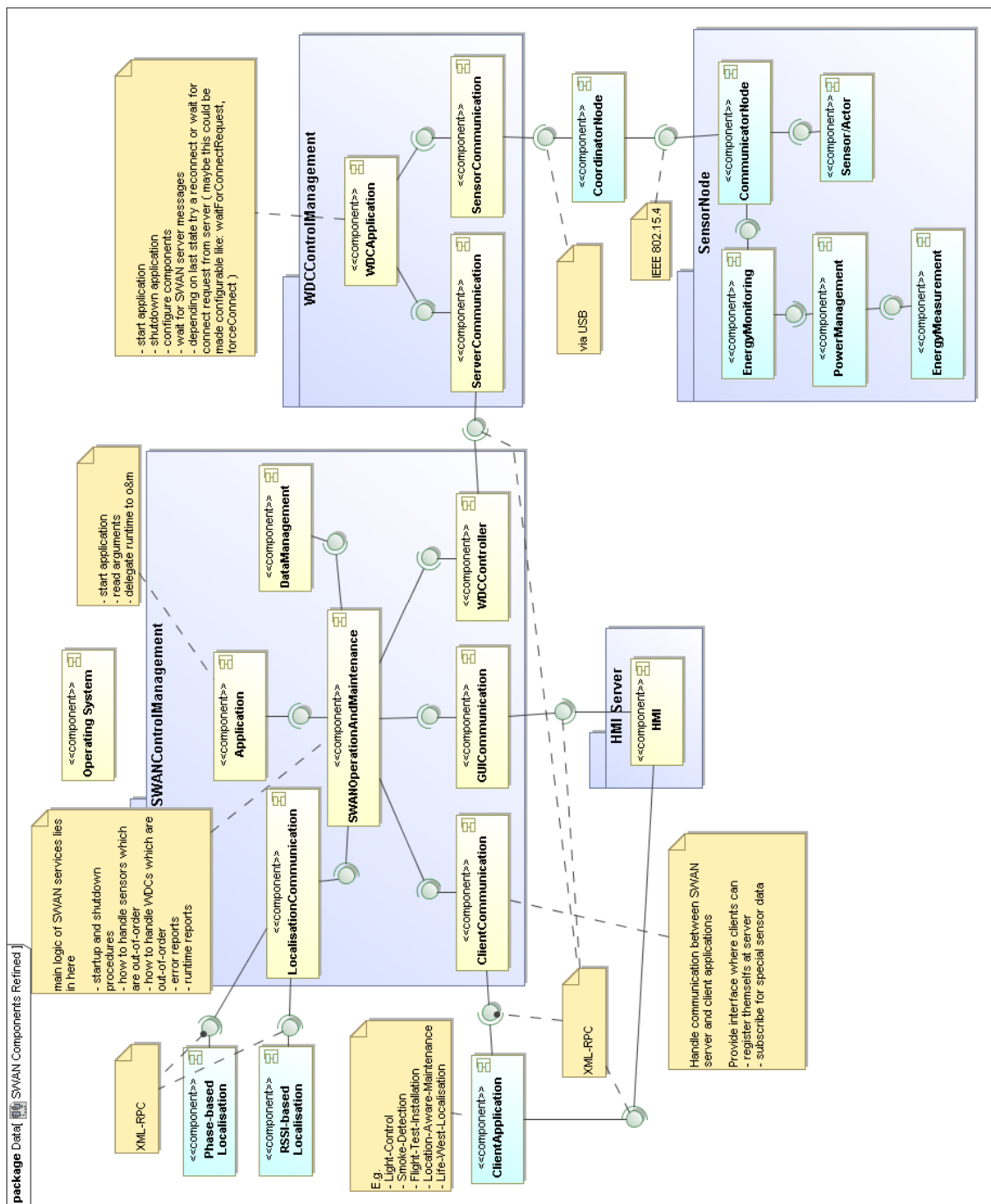


Abbildung 5: SWAN Software-Architektur

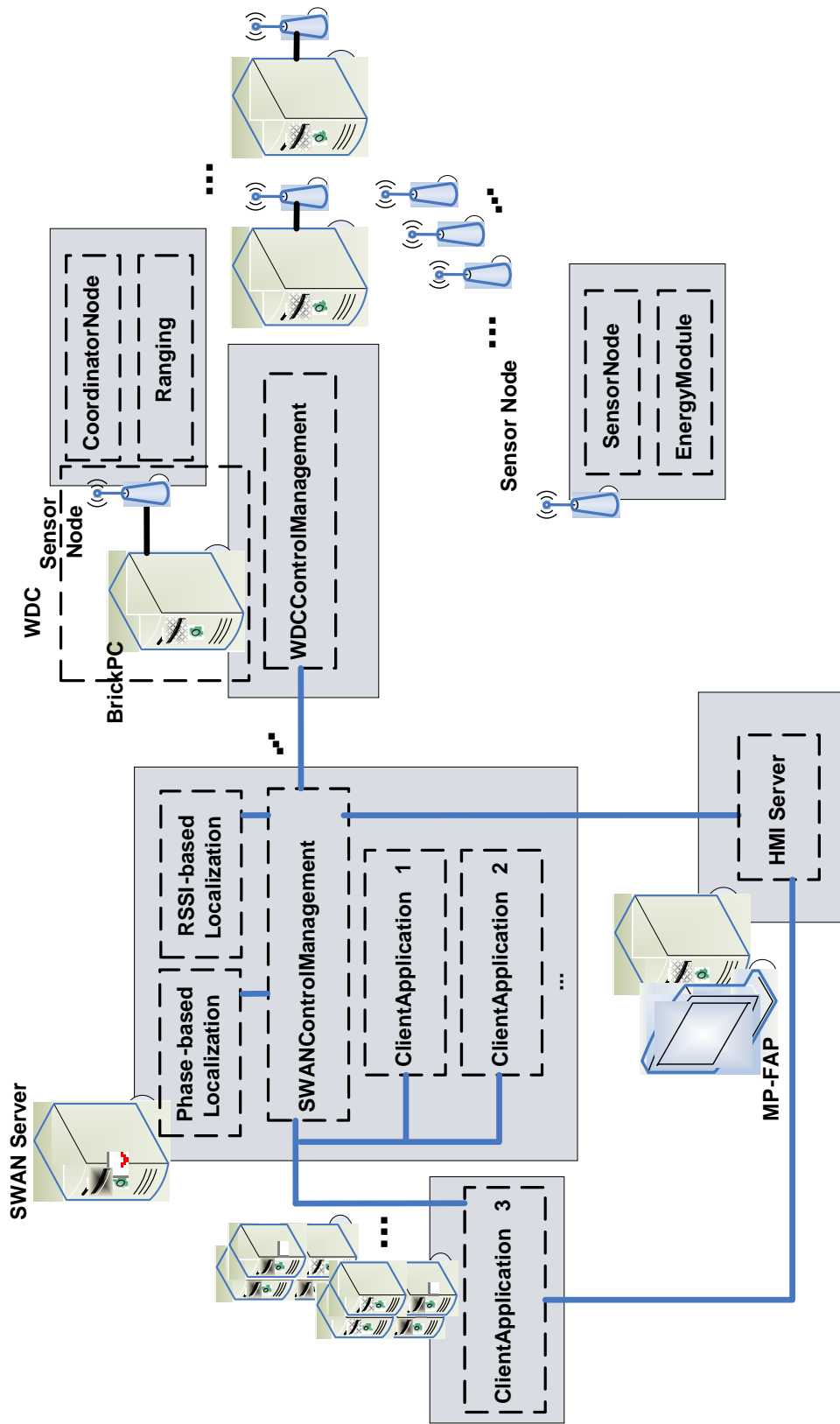


Abbildung 6: SWAN Funktionale Architektur mit verteilten Applikationen



Zum Verlauf der Arbeiten und den Iterationen hin zu dieser finalen Architektur sind noch folgende Anmerkungen zu machen:

- Festlegung der verschiedenen Software Schnittstellen-Protokolle:
  - XML-RPC zwischen Klienten, Server, Lokalisierung, HMI und WDC
  - SWAN-proprietär basierend auf RS232 zwischen SensorCommunication und CoordinatorNode
  - SWAN-proprietär basierend auf IEEE 802.15.4 zwischen CoordinatorNode und CoommunicatorNode, siehe Abbildung 5
- Definition, Erweiterung/Aktualisierung und finale Version der XML-RPC Schnittstelle zwischen:
  - ClientApplications (Temperatur, Licht, Fire/Smoke Detector und PMAT) und SWANControlManagement
  - Localization Clients (RSSI-basiert und Phase-basiert) und SWANControlManagement
  - HMI Server und und SWANControlManagement
- Spezifikation der zentralen Datenbank, siehe Abbildung 7
- Definition und Spezifikation von Verfahren und Regeln für die Operation des gesamten System in drei verschiedenen Zuständen: Initialisierung, normaler Betrieb und Wartung:
  - Registrierung von Klienten im System und Subskription am Knoten;
  - Gruppierung von Knoten durch die Klienten während der Initialisierung, um Broadcast-Nachrichten zu ermöglichen;
  - Routing-Regeln im Server, um die Kommunikation zwischen Klienten und angemeldeten Sensor-Knoten sicherzustellen und gleichzeitig die Kommunikation zu nicht angemeldeten Sensor Knoten zu vermeiden;
  - Identifikation von zeitkritischen und redundanz-erfordernden Applikationen;
  - Abläufe und Interaktionen zwischen Software-Komponenten für die zwei Lokalisierungsverfahren.
- Definition der XML-RPC Schnittstelle zwischen:
  - SWANControlManagement und WDCCControlManagement
  - WDCCControlManagement und CoordinatorNode
- Entfernung der in einer Zwischenversion enthalten EnergyControl Komponente vom SWAN Server entfernt wurde. Deren Funktionalität wird im Rahmen der Temperatur-Anwendung gezeigt.
- Vereinfachung der Spezifikation der zentralen Datenbank durch die Verlagerung von energiebezogener Information zu einer spezifischen Anwendung (Temperatur)
- Definition und Spezifikation von neuen nötigen Verfahren und Regeln für die Operation des gesamten Systems im Wartungs-Zustand, und zwar für:
  - Hinzufügen von neuen Knoten im System
  - Entfernung von Knoten aus dem System

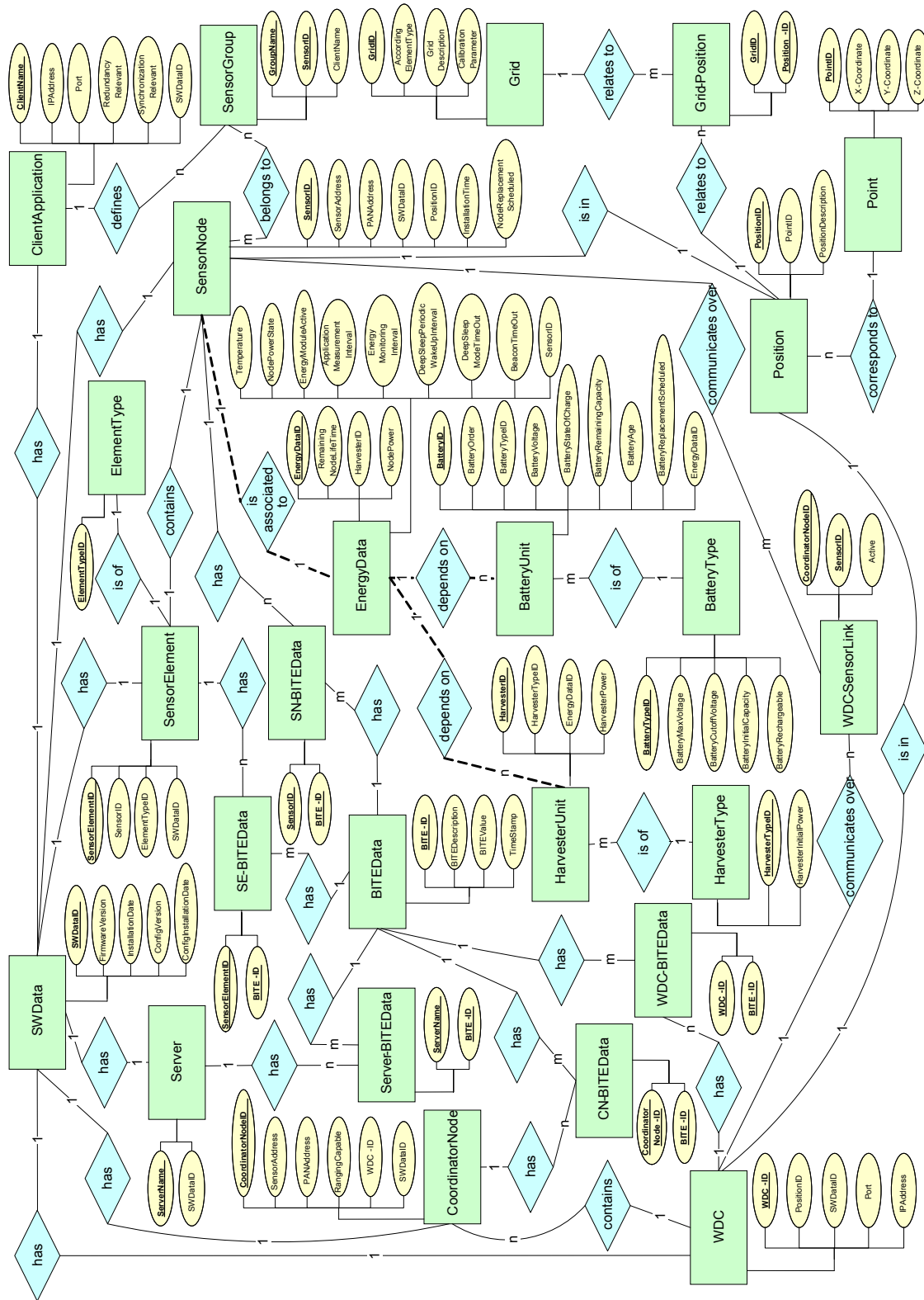


Abbildung 7: SWAN Datenbankmodell

- Spezifikation von Konfigurationsdateien für alle Anwendungen um den SWAN Server, um die Integrationsaufwände zu minimieren
- Erstellung von folgenden formellen ICDs (Interface Control Document), insbesondere für alle XML-RPC Schnittstellen um den SWAN Server
  - SWANControlManagement und ClientApplication. Dieses ICD gilt insbesondere für die Anwendungen, die für die Validations-Plattform ausgewählt wurden; die Spezifikation ist allerdings allgemeint formuliert und ist modular und einfach erweiterbar geblieben;
  - SWANControlManagement und Localisation Clients (RSSI-basiert und Phase-basiert);
  - SWANControlManagement und HMI Server;
  - WDCControlManagement und CoordinatorNode.

Diese ICDs wurden unter der Führung von TriaGnoSys erstellt und betreffen immer eine Komponente (des SWANControlManagement), die später von TriaGnoSys zu entwickeln war, und eine Komponente unter der Verantwortung eines weiteren SWAN-Partners.

- Vervollständigung der Spezifikation der SWANControlManagement Komponente als Zustandsmaschine, siehe Abbildung 8, mit den zugehörigen Regeln für die Zustandsübergänge, so wie die Definition von Abläufen und erlaubten/verbotenen Aktionen in jedem Zustand:
  - Zustand „INITIALIZATION“:
    - es wird geprüft, dass alle notwendige Konfigurations-Information in der zentralen Datenbank verfügbar ist;
    - es wird geprüft, ob alle WDCs erreichbar sind;
    - es wird geprüft, ob alle Sensorknoten verfügbar sind;
    - die Kommunikation zwischen SWANControlManagement und WDCControlManagements, und bis zu Sensorknoten darf stattfinden;
    - die Kommunikation zwischen SWANControlManagement und ClientApplications / HMI darf nicht stattfinden. Diese wird von den jeweiligen Kommunikations-Kontroller im SWANControlManagement blockiert;
  - Zustand „OPERATION“:
    - alle ClientApplications und LocalisationClients dürfen eine Verbingung mit dem SWAN Server herstellen, bzw. sich vom Netz trennen;
    - das SWANControlManagement informiert das HMI über die verfügbaren ClientApplications und LocalisationClients;
    - alle ClientApplications (außer LAMT) dürfen mit ihren jeweiligen (und nur mit denen!) Sensorknoten kommunizieren, sowie diese konfigurieren, Gruppen bilden, ihre eigene Wartungsaktionen durchführen, etc.;
    - alle ClientApplications (außer LAMT) dürfen ein RSSI-basiertes Lokalisierungsverfahren starten;
    - alle ClientApplications und das HMI dürfen Datenbank-Anfragen durchführen;

- das System wird in den MAINTENANCE Zustand wechseln nach Eingabe über das HMI;
- keine Anwendung darf ein Phase-basiertes- Lokalisierungsverfahren starten;
- Zustand „MAINTENANCE“:
  - die Kommunikation zwischen SWANControlManagement und ClientApplications darf nicht stattfinden. Daher auch nicht die Kommunikation zwischen ClientApplications und ihren Sensorknoten;
  - das System wird zu MAINTENANCE, LAMT-MODE, NODE-INSTALLATION oder OPERATION Zustand wechseln nach Eingabe über das HMI;
  - das HMI darf Datenbank-Anfragen durchführen;
- Zustand „LAMT-MODE“:
  - die LAMT ClientApplication ist die einzige ClientApplication, die mit dem SWANControlManagement kommunizieren darf;
  - die LAMT ClientApplication darf ein Phasen-basiertes Lokalisierungsverfahren starten;
  - das System wird zu MAINTENANCE Zustand wechseln nach Eingabe über das HMI; andere Anfragen von HMI werden in diesem Zustand nicht erlaubt;
- Zustand „NODE-INSTALLATION“:
  - die Kommunikation zwischen SWANControlManagement und ClientApplications darf nicht stattfinden. Daher auch nicht die Kommunikation zwischen ClientApplications und ihren Sensor Knoten;
  - die Kommunikation zwischen SWANControlManagement und dem HMI ist erlaubt;

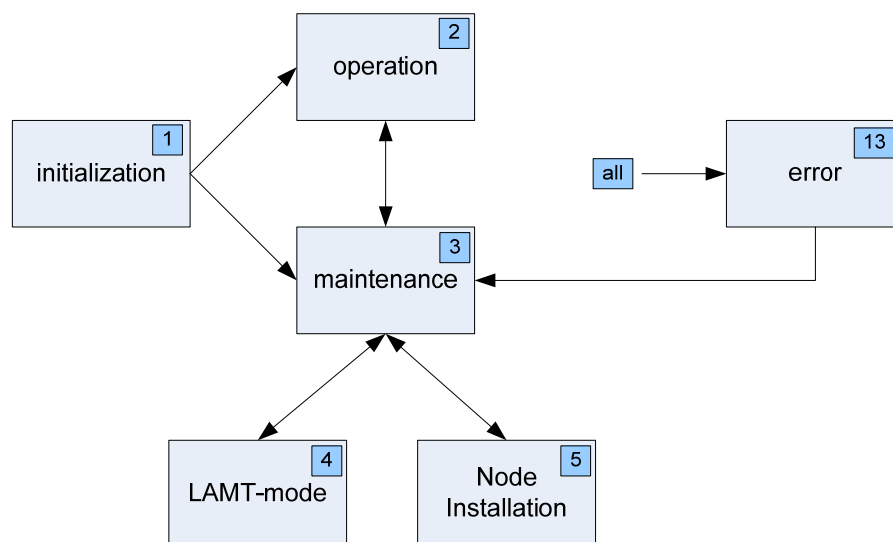


Abbildung 8: Zustände der SWANControlManagement Zustandsmaschine

### 2.1.3 AP 320: Lower Layer Node/Board Development

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

In AP320 erfolgt die WDC-Entwicklung verantwortlich durch TGS (Task322), wobei eine enge Abstimmung mit Partner EADS-IW an der Schnittstelle zu den Sensorknoten (Task321) erfolgt.]

Die ursprüngliche Planung für die Entwicklung eines Proof-of-Concept WDCs sah vor, den Koordinatorknoten an den BRIK PC für das Control Management über eine externe USB-Schnittstelle anzuschließen; dies in dem Wissen, dass Folgeprojekte wie CabWiSe sich um weitergehende Integration und Miniaturisierung der Knoten kümmern werden.

Um bereits in SWAN eine erste Stufe in Richtung „Integration“ zu gehen und damit ein etwas realitätsnäheres Design zu erzielen, wurden Möglichkeit untersucht, den Koordinatorknoten in den BRIK PC zu integrieren. Dies wurde letztlich mit zufrieden stellendem Ergebnis umgesetzt. Dazu wurden folgende Integrationsarbeiten ausgeführt:

- ein RCM Sensor Board wurde im Brik PC eingebaut, siehe Abbildung 9;
- dieses Sensor Board hat die Funktion eines Koordinators und kann eine oder zwei Antennen angeschlossen haben;
- die (Kommunikations-) Verbindung zwischen RCM Sensor Board und Brik PC erfolgt über eine interne USB Schnittstelle;
- das RCM Sensor Board ist so platziert, dass die beiden Antennen mit ausreichendem Abstand nach außen gezogen werden, um Antennen-Diversity zu ermöglichen; dies ist von Bedeutung für das Ranging im Rahmen der phasen-basierten Lokalisierung;
- mechanische Arbeiten am Brik PC zur Verbindung und Anbringung der Antennen am Gehäuse, siehe Abbildung 10

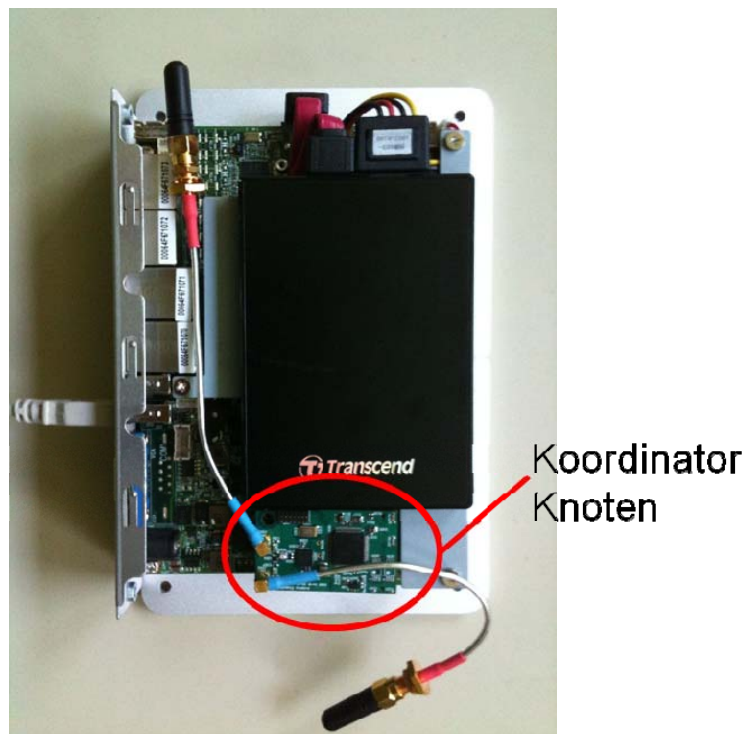


Abbildung 9: WDC Innenansicht mit einem Koordinator und zwei Antennen



**Abbildung 10: Außenansichten des integrierten WDC**

Die Endfertigung der Integration von Koordinatorknoten (RCM Sensor Board und USB Controller Board) in WDCs (BRIK PCs) wurde erfolgreich durchgeführt. Dafür wurde eine weitere Hardware-Schnittstelle am BRIK PC zu Verfügung gestellt, die zur Programmierung der nach innen gelegten Koordinatorknoten dient (die JTAG Schnittstelle, siehe Abbildung 10 links).

Vier integrierte WDC-Einheiten für die Validierungsplattform wurden an Airbus geliefert und dort ohne Notwendigkeit für weitere Modifikationen in verschiedenen Tests, Messungen und Demonstrationen erfolgreich eingesetzt.

#### **2.1.4 AP 330: Networking Software and Server Development**

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

TriaGnoSys übernimmt die Führung und weitgehende Implementierungsarbeiten im AP, wobei folgende spezifische Zuarbeit von Partnern vorgesehen ist:

- TUHH: Einbringung der Energiemanagementfunktionalität, Unterstützung der Implementierung;
- EADS-IW, Diehl, AOA: Unterstützung bei der Implementierung der Schnittstelle „Klientensoftware“ – „Systemsoftware“
- EADS-IW: Unterstützung bei der Implementierung der Schnittstelle „Kontinuierliche Erfassung eines mobilen Gerätes“
- Airbus: Implementierung HMI / Unterstützung bei der Implementierung der Schnittstelle „HMI Server“ – „SWAN Control Management Server“
- Airbus: Unterstützung bei der Implementierung der Schnittstelle „RSSI-based Lokalisierung“]

Gemäß der Spezifikation aus AP 310 wurden **Implementierungsarbeiten** zu folgenden Komponenten / Funktionalitäten auf dem **Server** durchgeführt.

- Konsolidierte Version mit Funktionen aller Komponenten (SWAN Application, SWANOperationAndMaintenance, ClientCommunication, DataManagement und WDCController) für alle fünf in Abbildung 8 dargestellten Zustände:



- *SWAN Application*  
Erstellt den Eintrittspunkt in das SWANControlManagement dar (enthält die main()-Funktion). Sie initialisiert die zentrale Komponente SWANOperationAndMaintenance und übergibt ihr die weitere Ablaufsteuerung. Zudem registriert sie eine Callback-Funktion beim Betriebssystem, die aufgerufen wird, sobald ein KILL, TERM oder INT (interrupt) Signal an die Applikation geschickt wird. Dadurch ist sie in der Lage, ein geordnetes Herunterfahren durchzuführen.
- *SWANOperationAndMaintenance*  
Erstellt jeweils eine Instanz der Komponenten DataManagement, ClientCommunication und WDCController. Diese werden in einer internen Registry abgelegt und können bei Bedarf aus ihr geholt werden. Um Nachrichten von Klienten effizient und transparent an die Sensoren weiterleiten zu können, wurde eine MessageQueue-Klasse implementiert. Eine Instanz von MessageQueue wird von SWANOperationAndMaintenance erstellt und an die Komponenten ClientCommunication und WDCController weitergegeben. ClientCommunication schreibt Nachrichten in diese Queue und WDCController liest sie aus. Durch diese Vorgehensweise wird eine angemessene Kommunikation zwischen den beiden Komponenten ermöglicht.
- *ClientCommunication*  
Stellt über einen RPC-Server die spezifizierten Schnittstellen-Funktionen den Klienten-Applikationen zur Verfügung, tritt in der Rolle des Servers auf. Sie reagiert auf Anfragen von Klienten, indem sie entweder Nachrichten an Sensoren in die MessageQueue schreibt, oder, wenn Sensor-Informationen angefordert werden, diese beim DataManagement einholt und an den anfragenden Klienten zurück gibt. Desweiteren können Klienten sich am SWAN Netz registrieren, abmelden und sich für Nachrichten von Sensoren subscriben.
- *DataManagement*  
Das DataManagement dient als Abstraktionsschicht zwischen der Datenhaltung und der Datenverwendung. Alle im SWAN Network beteiligten Komponenten kennen lediglich die Schnittstelle, die das DataManagement zur Verfügung stellt und brauchen sich keine Gedanken um Datenbank-Systeme, Tabellen und dergleichen zu machen. Derzeit können Sensordaten mit Positions-, Typ- und Address-Information den Klienten zur Verfügung gestellt werden.
- *WDCController*  
Der WDCController wartet auf Nachrichten in der MessageQueue, liest sie aus und gibt sie über eine XML-RPC Schnittstelle an das WDCControlManagement weiter. Der WDCController hat in diesem Fall die Rolle des Klienten gegenüber dem WDCControlManagement.
- Eine endgültige Version der zentralen SWAN-Datenbank wurde in mehreren Iterationen erstellt und in das SWANControlManagement integriert. Es existieren SQL-Skripte zum Erstellen der Datenbank-Tabellen, zum Befüllen mit Test-Daten und für die komplexen Anfragen der verschiedenen Operationen des Servers.

Gemäß der Spezifikation aus AP 310 wurden **Implementierungsarbeiten** zu folgenden Komponenten / Funktionalitäten auf dem **WDC** durchgeführt.

- Eine konsolidierte Version mit kompletten Funktionalitäten aller Komponenten (WDCApplication, ServerCommunication und SensorCommunication) für alle fünf in Abbildung 8 dargestellten Zustände:
  - *WDCApplication*  
Von der Funktionalität her äquivalent zur Komponente SWAN Application.
  - *ServerCommunication*  
Von der Funktionalität her äquivalent zur Komponente ClientCommunication.
  - *SensorCommunication*  
Hier wird die serielle Kommunikation mit dem CoordinatorNode realisiert.

### 2.1.5 AP 340: Application/Client Software Development

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

In AP340 entwickelt TGS eine generische Klientensoftware, die soweit möglich auf die spezifischen Anwendungen der Partner AOA, Diehl und EADS-IW abgestimmt ist.]

Die ursprünglichen Pläne zum Entwurf einer umfassenden generischen Klientensoftware (i) als Basis für die weitere Entwicklung spezifischer Klienten sowie (ii) für umfangreiche Integrationstests wurden in Abstimmung zwischen den Verbundpartnern reduziert und fokussiert.

Wegen der Komplexität der Gesamtintegration im Verbund erschien es in einer ersten Iteration weckmäßig, auf Basis eines einfachen Temperatur-Sensors eine entsprechende Applikation zur Temperaturüberwachung mit begrenzter Funktionalität zu erstellen, um systematisch prinzipielle Tests zur Server- und WDC-Funktionalität durchführen zu können.

Dazu wurden insbesondere folgende Implementierungsarbeiten durchgeführt:

- Applikation auf einem Sensorknoten mit drei verschiedenen Modi: request, threshold und periodic;
- Klienten-Applikation, die mit dem o.g. Sensorknoten kommuniziert für Steuerung, Konfiguration und Erfassung/Überwachung von Temperaturwerten.

In nächsten Zyklen der Softwareentwicklung wurde Beispiel-Code für die XML-RPC Schnittstelle der Sub-Komponenten ClientCommunication, LocalisationCommunication und GUICommunication des SWANControlManagements bei TriaGnoSys entwickelt, getestet und dann den Partnern zur Verfügung gestellt. Das Ziel dieser Maßnahme war eine sicherere und effiziente Integration unter verschärften Terminbedingungen in der finalen Phase des Projekts.



### 2.1.6 AP 350: Localisation

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

In AP350 stellt TGS sicher, dass die von Partnern realisierten Lokalisierungsfunktionen und -Algorithmen serverseitig unterstützt werden und unterstützt bei der Integration dieser Funktionen auf dem Server; es ergibt sich eine klare Softwareschnittstelle insbesondere zum Maintenance-PDA-Klienten.]

Wesentliche Aufgabe war hier die Definition und Spezifikation von Regeln für die Behandlung von Lokalisierungsanfragen abhängig vom Systemzustand (normaler Betrieb oder Wartung).

Hintergrund, Situation:

- Das phasen-basierte Lokalisierungsverfahren benötigt die gesamte Bandbreite auf dem drahtlosen Link und blockiert die Kommunikation zwischen Klienten und ihren Knoten.
- Das RSSI-basierte Lokalisierungsverfahren ist zeitaufwendig und könnte die Kommunikation zwischen Klienten und ihren Knoten kritisch verzögern.

Entscheidungen:

- Eine Lokalisierung wird beim SWAN-Server nur im Wartungs-Modus erlaubt (wenn die Klienten-Knoten-Kommunikation nicht kritisch ist).
- Innerhalb des Wartungs-Modus gibt es ein Unterzustand „LAMT-Modus“, in dem die kontinuierliche Positions-Erfassung eines mobilen Gerätes möglich ist (phasen-basierte Lokalisierung)

Regeln:

- Eine Lokalisierungs-Anfrage im normalen Betrieb (durch HMI oder Klienten) wird beantwortet mit den gespeicherten Werten in der Datenbank.
- Eine Lokalisierungs-Anfrage im Wartungs-Modus (durch HMI oder Klienten) triggert die RSSI-basierte Lokalisierung.
- Eine Lokalisierungs-Anfrage im LAMT-Modus (durch den LAMT-Klienten oder von der HMI-Seite des LAMT) triggert die phasen-basierte Lokalisierung.

Der detaillierte Ablauf ist dem Flussdiagramm in Abbildung 11 zu entnehmen.

Die Regeln für die Behandlung von Lokalisierungsanfragen abhängig vom Systemzustand (normaler Betrieb oder Wartung) wurden von TriaGnoSys geschrieben. Die Vervollständigung dieser Spezifikation wurde im nächsten Schritt durch den Entwurf von Ablaufdiagrammen für alle Fälle und System-Zustände erreicht, wie in Abbildung 12 bis Abbildung 16 zusammenfassend dargestellt.

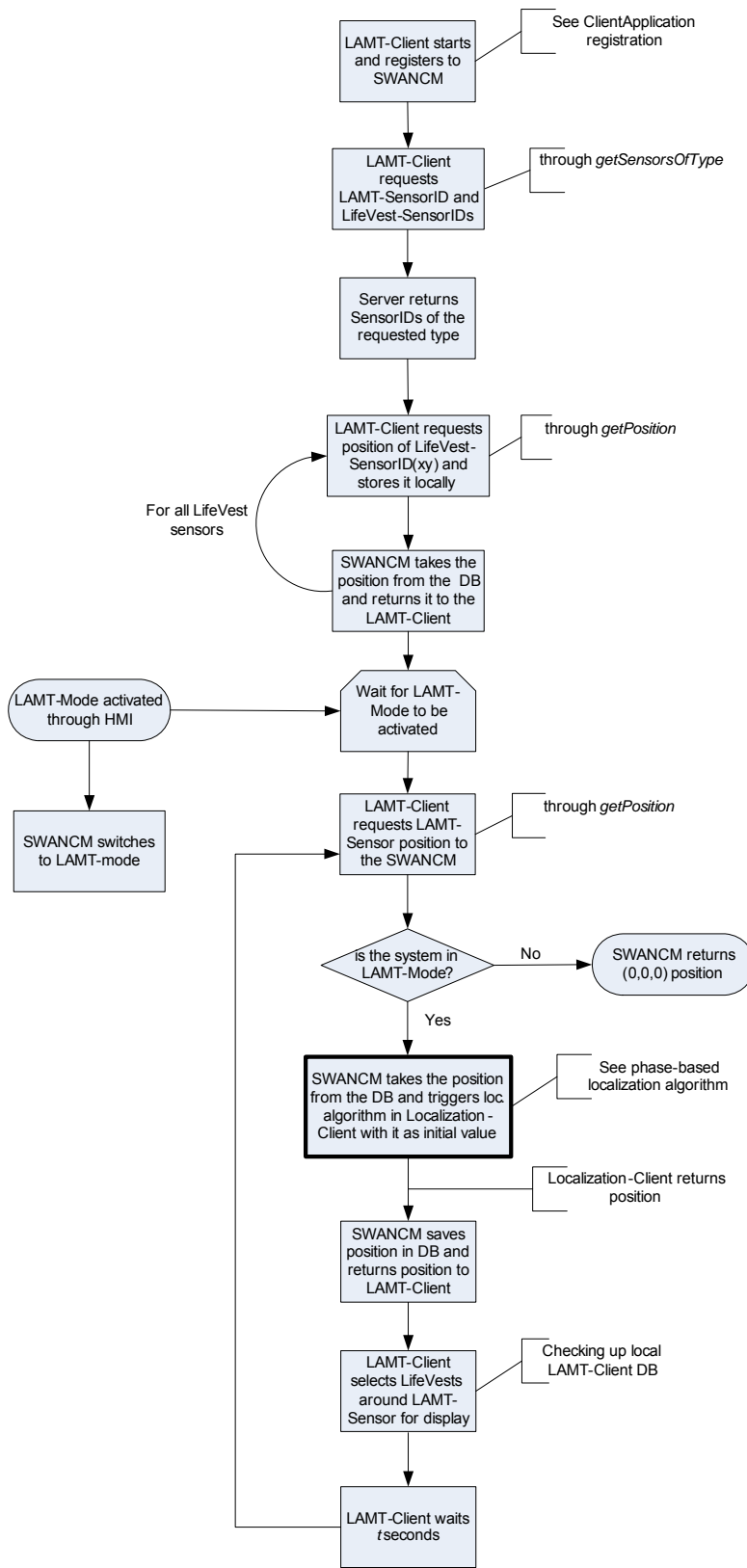


Abbildung 11: Flussdiagramm zur Lokalisierung

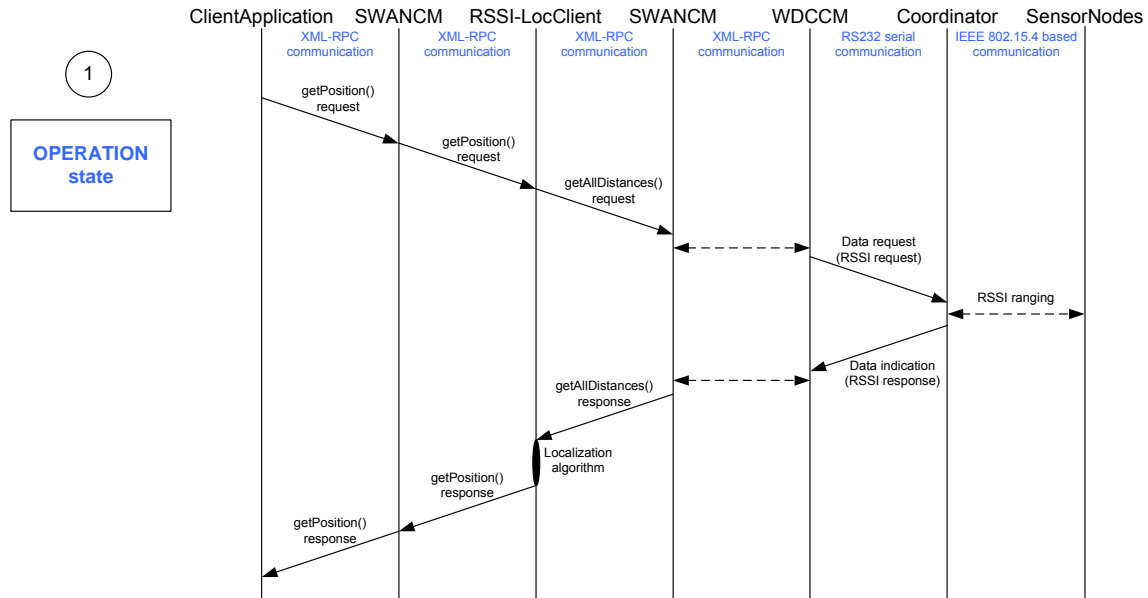


Abbildung 12: Ablaufdiagramm einer Lokalisierungsanfrage von einer ClientApplication im normalen Operations-Zustand

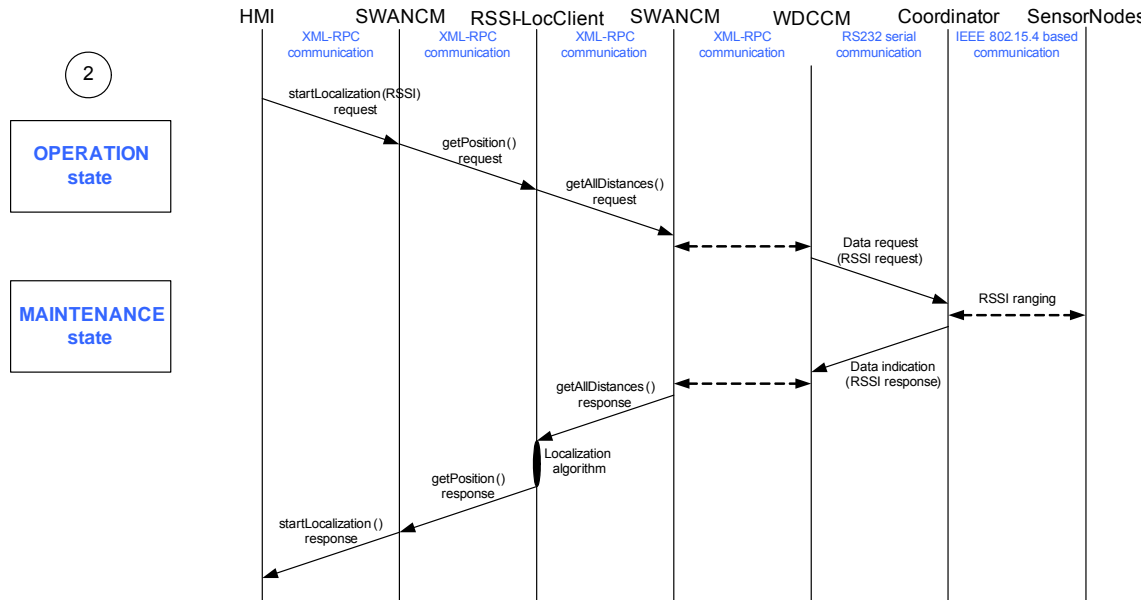


Abbildung 13: Ablaufdiagramm einer RSSI-Lokalisierungsanfrage des HMIs im normalen Operations- oder Wartungs-Zustand

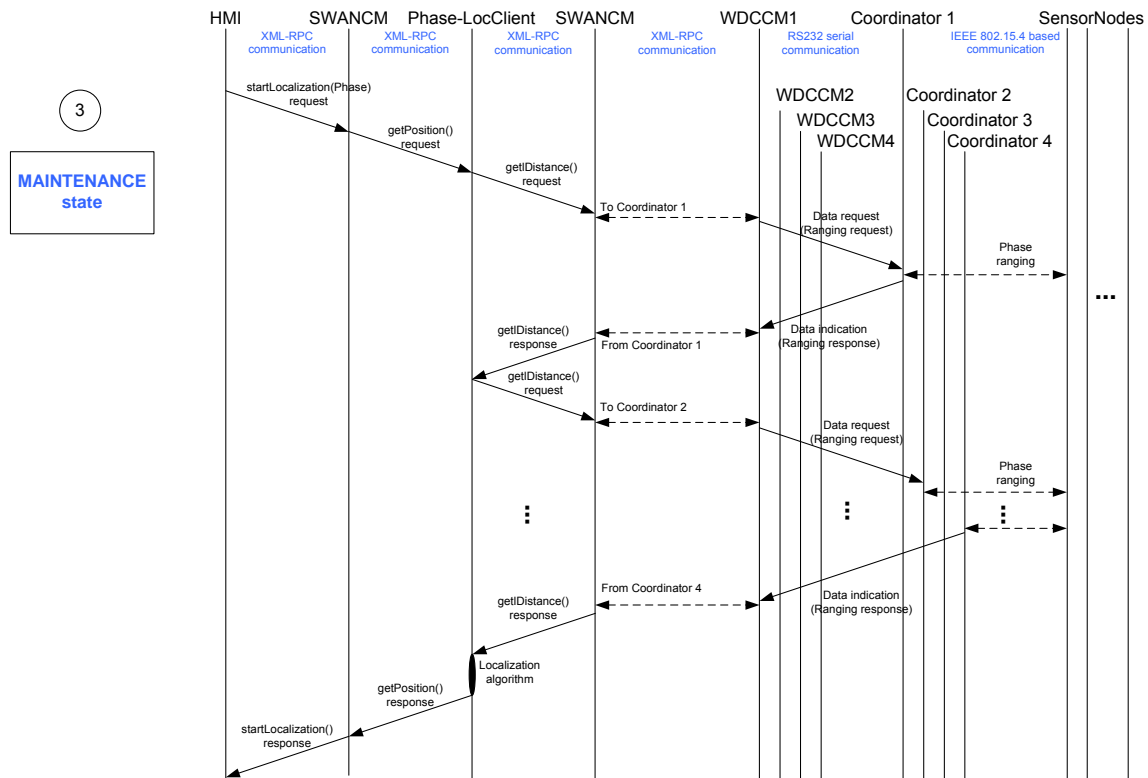


Abbildung 14: Ablaufdiagramm einer Phase-based-Lokalisierungsanfrage des HMIs im Wartungs-Zustand

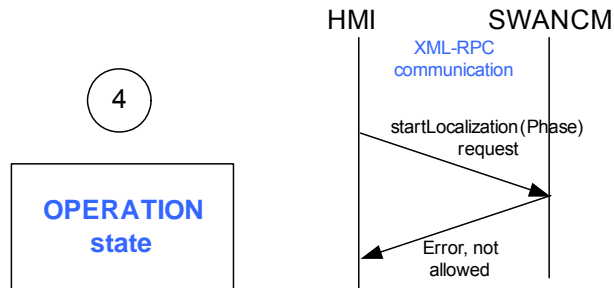
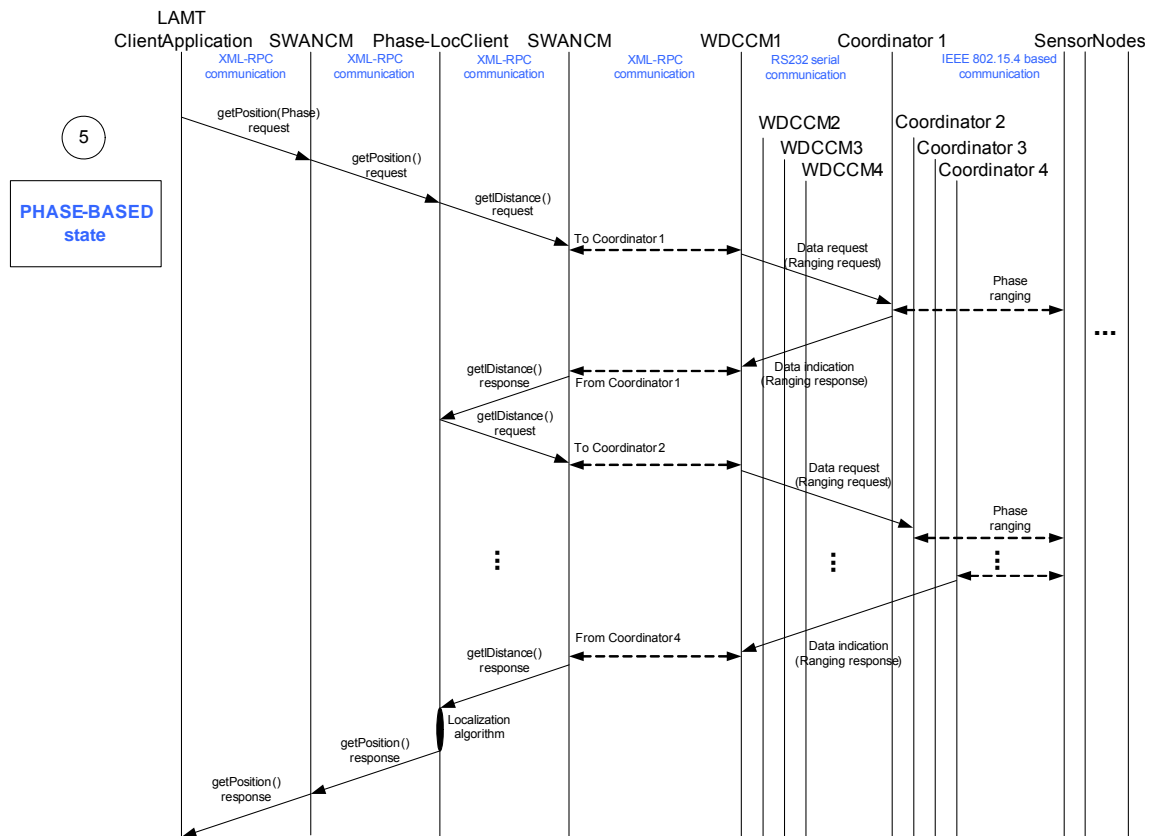


Abbildung 15: Ablaufdiagramm einer Phase-based-Lokalisierungsanfrage des HMIs im normalen Operations-Zustand



**Abbildung 16: Ursprüngliches Ablaufdiagramm einer Phase-based-Lokalisierungsanfrage der LAMT Anwendung im normalen Phase-based-Zustand**

Auf Basis der Erfahrungen aus ersten Implentierungen und Tests wurden im letzten Zyklus der Arbeiten einige Verbesserungen und Anpassungen in den Abläufen der beiden Lokalisierungsverfahren nötig:

- Keines der beiden Verfahren darf durch das HMI initiiert werden;
- RSSI-basierte Lokalisierung: Während das Ablaufdiagramm des RSSI-basierten Lokalisierungsverfahren unverändert bleibt, erfolgt die Messung von RSSI-Ranging Werten nun nicht mehr durch getriggerte Messungen in zeitlichen Abständen; stattdessen läuft diese Messung zwischen den betroffenen Sensor Knoten ständig im Hintergrund. Dafür musste die Schnittstelle zwischen WDCControlManagement und CoordinatorNode angepasst werden;
- Phasen-basierte Lokalisierung: Der neue Ablaufdiagramm dieses Verfahrens ist in der Abbildung 16 dargestellt. Die kontinuierliche Anfrage an den Phase-based LocalisationClient wird vom SWANControlManagement gesteuert und nicht mehr von der LAMT ClientApplication. Außerdem werden die Anfragen an die verschiedenen WDCControlManagement-Instanzen parallel gesendet und nicht mehr hintereinander. Dadurch wird eine bessere Phasenmessung erzielt.

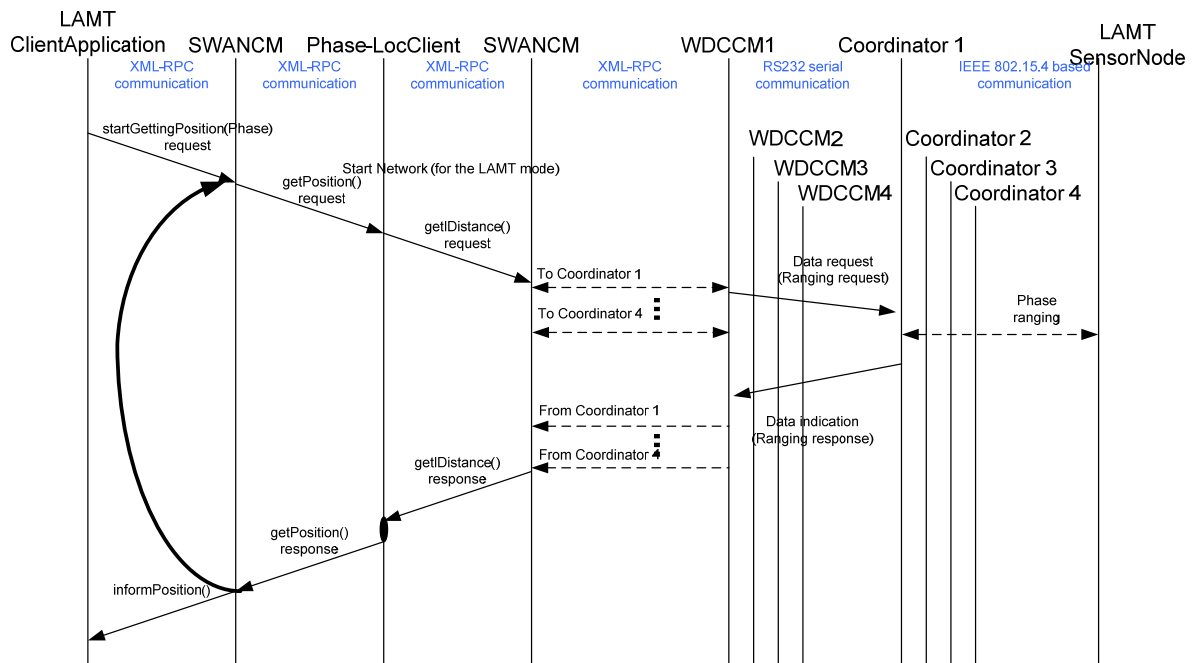


Abbildung 17: Verbessertes Ablaufdiagramm einer Phase-based-Lokalisierungsanfrage der LAMT Anwendung im normalen Phase-based-Zustand

### 2.1.7 AP 410: Test Cases

Das Arbeitspaket wurde mit Zuarbeit von TriaGnoSys als Unterauftragnehmer von Airbus bereits vor dem formellen Vertragsbeginn des umgestellten Vorhabens abgeschlossen.

### 2.1.8 AP 420: Integration of Validation Platform

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

TGS führt die Integration aller Soft- und Hardwareanteile zu einer Validationsplattform in AP420 und arbeitet dabei in mehreren klar definierten Phasen mit allen Verbundpartnern zusammen.]

Die **erste Software-Integrationsphase** wurde mit initialen Versionen der verschiedenen Software-Komponenten durchgeführt. Insbesondere wurde der Fokus zunächst auf die Schnittstellen zwischen den Komponenten gelegt und danach auf den automatischen Ablauf der spezifizierten Prozesse (siehe unten).

Folgende Schnittstellen wurden in der ersten Integrationsphase erfolgreich integriert:

- WDCControlManager und CoordinatorNode (außer Lokalisierungs-Nachrichten, die für die zweite Phase vorgesehen sind)
- SWANControlManager und CoordinatorNode (ebenso ohne Lokalisierungs-Nachrichten)
- SWANControlManager und Licht-ClientApplication, zu 100%
- SWANControlManager und Fire/Smoke Detector-ClientApplication, zu 100%
- SWANControlManager und RSSI-Lokalisierungsklient

- SWANControlManager und Phase-based-Lokalisierungsklient

Folgende Prozesse, die die Interaktion von mehreren Software-Komponente fordern, wurden getestet:

- Anmeldung von vordefinierten Sensoren am System, sowie Ablehnung von solchen Anmeldungen wenn die Sensoren im Server nicht bekannt sind
- Registrierung von ClientApplications am Server nach Initialisierung
- Anmeldung von ClientApplications an Sensoren, um Nachrichten mit ihnen austauschen zu dürfen
- Gruppierung von Sensoren bei ClientApplications, um Broadcast-Nachrichten senden zu dürfen

Der gesamte Integrationsprozess dieser ersten Phase wurde in entsprechend angemessenen Zyklen von eigentlichen Integrationsarbeiten, Test&Verifikation, Debugging/Fehlersuche und Fehlerbehebung umgesetzt.

Ein zentraler Schritt zum Erfolg des gesamten Projekts waren hierbei war die erfolgreiche Datenübertragung zwischen zentralem Server und den Sensorknoten mittels WDC und Koordinatorknoten, sowie die komplette fehlerlose Kommunikationskette zwischen der Fire/Smoke Detector ClientApplication und einem Fire/Smoke Detector Sensorknoten.

Die **zweite und dritte Integrationsphase** wurden im TriaGnoSys *Integration Laboratory* durchgeführt (siehe auch Sektion 2.1.9 / AP 440), und eine letzte Integrationsaktion wurde in einem Flugzeug-Mockup bei Airbus in Hamburg unter TriaGnoSys-Führung durchgeführt.

Wie in der ersten Phase wurde der gesamte Integrationsprozess in entsprechend angemessenen Zyklen von eigentlichen Integrationsarbeiten, Test&Verifikation, Debugging/Fehlersuche und Fehlerbehebung umgesetzt.

In **der zweiten Integrationsphase** wurde der Fokus auf folgenden Schnittstellen bzw. Prozesse gelegt:

- SWANControlManager und HMI;
- SWANControlManager und Temperature ClientApplication, mit grundlegenden Funktionalitäten;
- SWANControlManager und Life-vest ClientApplication, mit grundlegenden Funktionalitäten;
- Alle ClientApplications und HMI;
- RSSI-based Lokalisierungsabläufe ab den SWANControlManager bis auf die FTI Sensor Knoten;
- Phase-based Lokalisierungsabläufe ab den SWANControlManager bis auf die LAMT Sensor Knoten.

Außerdem wurden aktualisierte Versionen der Software-Komponenten aus der ersten Integrationsphase fortlaufend im Testnetz installiert, neu integriert und getestet.

In der **dritten Integrationsphase** wurden die letzten Funktionalitäten der verschiedenen Anwendungen integriert, sowie die Schnittstelle zwischen SWANControlManager und LAMT ClientApplication. Insbesondere wurde für die Temperature ClientApplication die Nachrichten über Energie-bezogene Information auf den Sensorknoten mitgeliefert und entsprechend integriert, und für die Life-vest ClientApplication der Sensorknoten Zustand in der ClientApplication bearbeitet. Nebenbei wurden die zwei Lokalisierungsabläufe in dieser

Phase von den jeweiligen ClientApplications angestoßen (Temperature ClientApplication für die RSSI-based Lokalisierung und LAMT ClientApplication für die Phase-based Lokalisierung). Es ist erwähnenswert, dass die erhaltenen Werte von beiden Lokalisierungsverfahren nicht im TriaGnoSys Labor verifiziert werden konnten, da dafür eine bestimmte Anpassung der Kalibrierung des Algorithmus (für einen Raum bestimmt) nötig gewesen wäre. Die verfügbare Kalibrierung war für die Umgebung im End-System (Airbus Mockup) geeignet und wurde vom Projektpartner EADS-IW bereitgestellt.

Ein zentraler Schritt zum Erfolg dieser letzten Integrationsphase im TriaGnoSys-Labor waren die dedizierten Arbeitssitzungen mit jedem Partner. In einer dieser Arbeitssitzungen mit Diehl Avionik wurde entschieden, dass die Wartungsaktionen für die Licht-Steuerung nicht im End-System mit gezeigt werden sollen, da die Größe der dafür nötigen Kommunikationsnachrichten Probleme an der Luft-Schnittstelle erwarten ließen.

Zusätzlich wurden Systemtests durchgeführt, in denen alle Anwendungen eingeschaltet waren und im Parallelbetrieb getestet wurden. Bei diesen Systemtests wurden die (momentanen) Grenzen der Skalierbarkeit hinsichtlich der Performance (hohe Anzahl von Nachrichten in kurzer Zeit) vor allem an der Schnittstelle zwischen WDCControlManagement und Koordinatorknoten ausgelotet und als wichtige Erfahrung für zukünftige Arbeit verbucht.

Eine letzte **Integrationsitzung** fand im **Flugzeug-Mockup** bei Airbus vor der finalen Demonstration mit alle beteiligten Partnern statt. Als Zuständiger für die gesamte Integration erstellte TriaGnoSys das Dokument „SWAN Final Demo Network Setup“, in dem alle Konfigurationsparameter (Sensor Knoten Adressen, XML-RPC Schnittstellen-Parameter, Software Information, etc.) und Einrichtung (Positionsinformation, Netzeinrichtung, etc.) aller SWAN System Komponenten gesammelt wurden, einschließlich Richtlinien für die Konfiguration der zentralen Datenbank. Abbildung 18 zeigt die Netz-Einrichtung der SWAN Validations-Plattform für das Mockup bei Airbus.

Neben der Software-Integration wurde im Mockup (Abbildung 19) die zusätzliche Installation und das Zusammenspiel aller Hardware-Einheiten durchgeführt und verifiziert, die während der Test&Lab Trials (AP 440) nicht unbedingt gleichzeitig bei TGS verfügbar waren:

- FTI Knoten – 9 Einheiten, siehe Abbildung 20 (a)
- Life-vest Knoten – 9 Einheiten, siehe Abbildung 20 (a)
- Lichtleisten – 9 Einheiten, siehe Abbildung 20 (b)
- Rauchmelder – 4 Einheiten, siehe Abbildung 20 (c)
- WDC – 4 Einheiten, siehe Abbildung 20 (d)



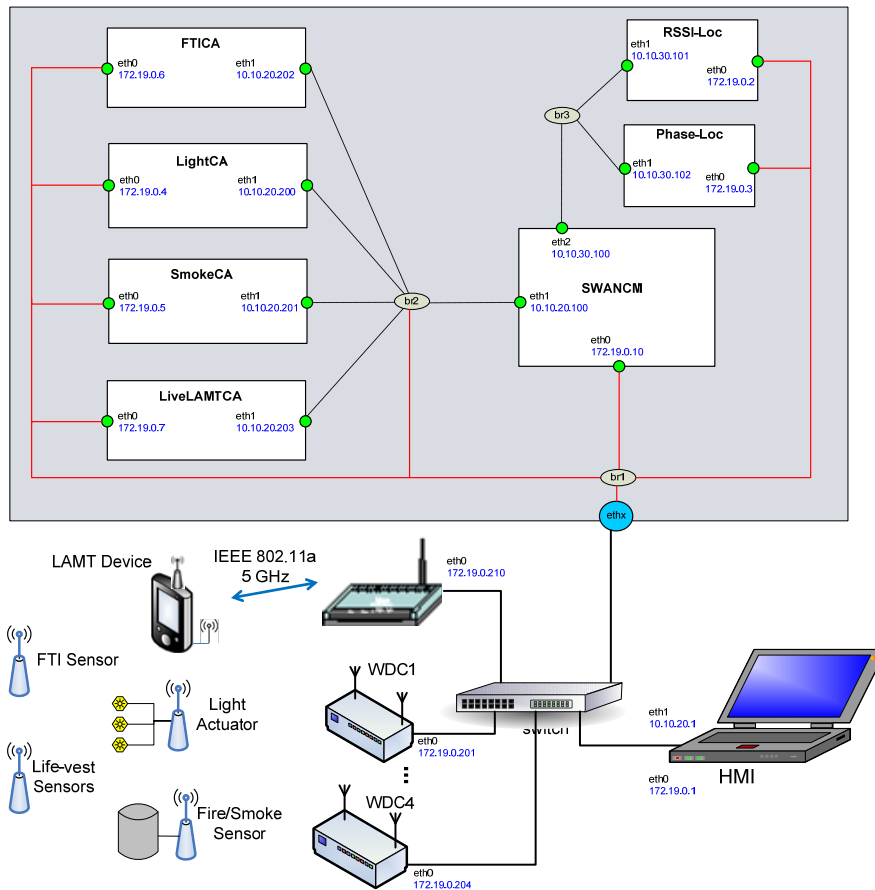


Abbildung 18: Netz-Einrichtung der SWAN Validations-Plattform

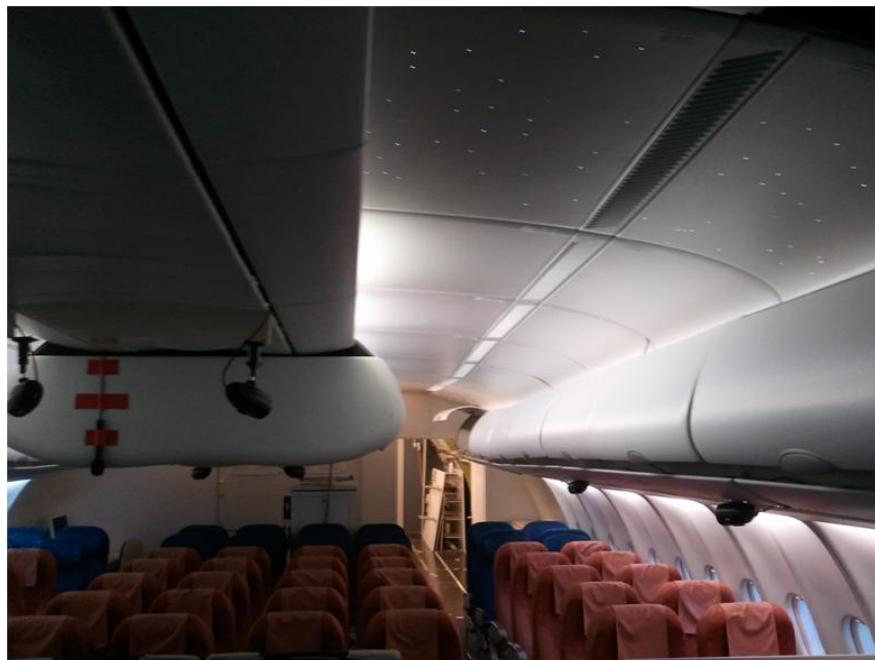
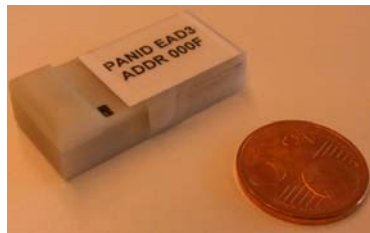
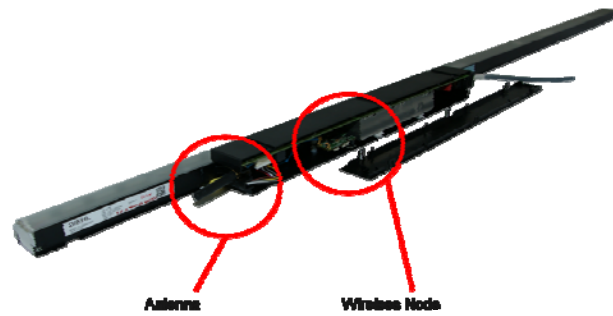


Abbildung 19: Flugzeug-Mockup bei Airbus



(a) Sensorknoten



(b) Licht-Leiste mit integriertem Sensorknoten



(c) Rauchmelder mit integriertem Sensorknoten



(d) WDC und interne Sicht der Koordinator-Knoten

Abbildung 20: Hardware der SWAN Validations-Plattform

### 2.1.9 AP 440: Test&Lab Trials

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

TGS führt in AP440 verantwortlich alle begleitenden Geräte- und Systemtests sowie Trials in der Integrationsumgebung durch, die sich wiederum eng an die Arbeiten in AP420 anlehnen und damit auf der kontinuierlichen Abstimmung mit den jeweiligen Partnern beruhen.]

Für den effizienten und gleichzeitig revisionskontrollierten Austausch und die Pflege der gesamten Software im Projekt wurde von TriaGnoSys ein Repository eingerichtet und allen beteiligten Partnern im Projekt per geschütztem Remote-Zugang zur Verfügung gestellt.

Im TriaGnoSys Integration&Test Labor wurde ein SWAN Testnetz eingerichtet mit einem Server, der 7 virtuelle Maschinen hostet, siehe grauer Block in Abbildung 21. Jede SWAN-Komponente läuft auf je einer dieser virtuellen Maschinen. Zum Testnetz gehören auch 4 WDCs und mehrere Sensorknoten. Zusätzlich gibt es eine externe Verbindung zum TriaGnoSys-Intranet für Steuerung des Testnetzes, Durchführung von Tests, und für Debugging-Zwecke.

In einer nächsten Phase wurde das SWAN Testnetz im TriaGnoSys SIL (System Integration Laboratory) erweitert, um alle Anwendungen, die für die SWAN Validations-Plattform vorgesehen waren, *hosten* zu können, siehe Abbildung 21. Das eingerichtete TriaGnoSys SIL entspricht einer reduzierten Version der SWAN-Validations-Plattform, mit folgenden Komponenten:

- 1 Rauchmelder Sensor Knoten (4 in der Validations-Plattform)
- 2 Licht-Aktuatoren (9 in der Validations-Plattform)
- 2 Temperatur Sensor Knoten (9 in der Validations-Plattform)
- 2 Life-Vest Sensor Knoten (9 in der Validations-Plattform)
- 1 LAMT Sensor Knoten, nicht an einem LAMT-Gerät angeschlossen (1 in der Validations-Plattform)
- 4 WDCs (4 in der Validations-Plattform)

Die einzigen Komponenten, die im TriaGnoSys SIL nicht integriert wurden, waren das LAMT-Gerät und die 5 GHz WLAN AP für die Kommunikation zwischen dem LAMT-Gerät selber und der LAMT ClientApplication, da dafür keine Integrationsarbeit zwischen verschiedenen SWAN-Partnern erforderlich war.

Eine zusätzlicher Unterschied zwischen dem SWAN TriaGnoSys SIL und der SWAN Validationsplattform wird aus der Vergleich von Abbildung 21 mit Abbildung 18 deutlich: das HMI ist im TriaGnoSys SIL auf einer virtuelle Maschine installiert, während in der Validationsplattform ein dedizierter physikalischer Rechner dafür zur Verfügung steht.

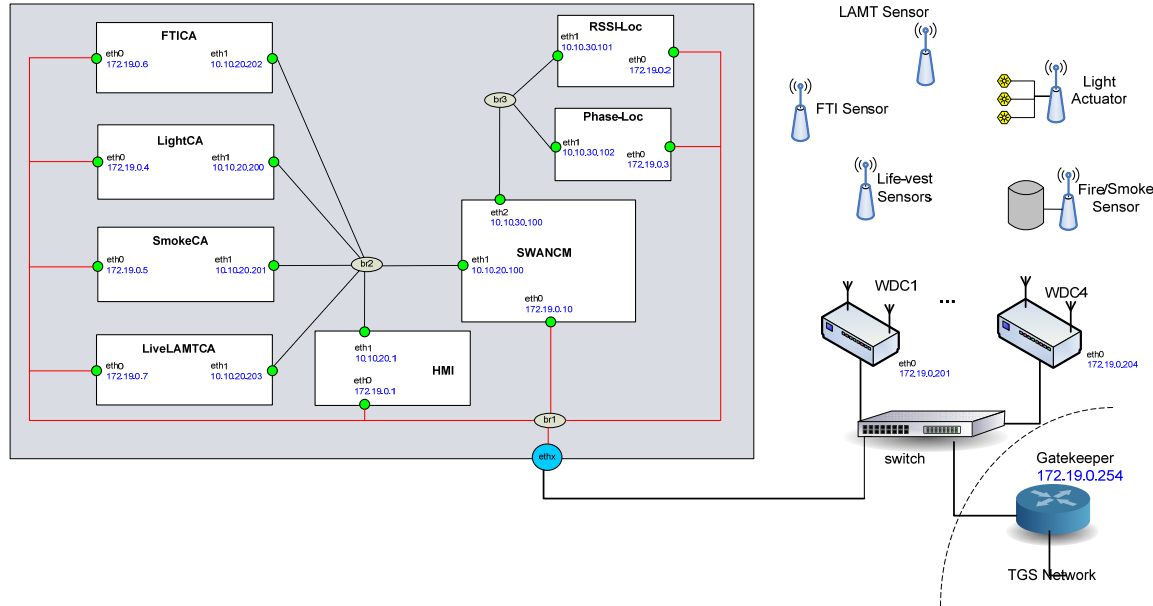


Abbildung 21: SWAN Testnetz im TriaGnoSys-Labor

## 2.2 Detaillierte Darstellung der Ergebnisse – Teilprojekt SO4A

### 2.2.1 AP 131: Funktionale Architektur

Die Zuarbeiten von TriaGnoSys zu diesem Airbus-geführten AP wurden bereits vor dem formellen Vertragsbeginn des umgestellten Vorhabens abgeschlossen.

### 2.2.2 AP 132: Funktionale Architektur für Kommunikationsinfrastruktur

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

TGS führt die Definition der Kommunikationsinfrastruktur für eine Service-orientierte Architektur und wird dabei von Airbus und TU Ilmenau unterstützt.]

Die zentrale Aufgabe war die Spezifikation der SO4A Kommunikations-Testbed-Umgebung mit dem Fokus der TGS-Arbeiten auf den Air-Ground-Link.

Zusätzlich zur ursprünglichen Planung des APs im Verbundvorhaben erfolgte wegen der TGS-spezifischen Erweiterung und neuen Verwertungsplänen insbesondere auch eine erste Anpassung und Abgleich der Architekturpläne mit anderen laufenden und geplanten TGS-Arbeiten/-Projekten vor allem im Bereich der SWIM-Ansätze (System Wide Information Management) im Bereich der zukünftigen Flugverkehrsführung, welche Ähnlichkeiten und Synergien mit den SoA-Ansätzen innerhalb von @MOST-SO4A aufweist.

Abbildung 22 zeigt die SO4A-Kommunikationsarchitektur; die Abgrenzung zwischen den im SO4A-Projekt betrachteten Wartungsdiensten und Wartungsaspekten von den SWIM-Diensten zur Flugverkehrsführung wird bei aller Ähnlichkeit der Ansätze deutlich.

Eine Reihe von erforderlichen System-/Softwarekomponenten, Software-Applikationen, Beispieldiensten und Nachrichten wurden identifiziert und ihre Funktionalitäten wurden festgelegt:

- Software-Komponenten: Airborne Broker, Airborne Peer, Ground Broker, Ground Peer
- Software Applikationen und Dienste: Link Controller, Klient oder Dienst (z.B. Meldungen über fehlerhafte Teile, Wetter-Dienst)
- Nachrichten: Broker Subscription, Notification, Publish

Auf dieser Basis wurden die Interaktionen zwischen den verschiedenen SW-Komponenten und -Applikationen unter Nutzung von Use Cases spezifiziert, und es wurden effiziente Übertragungsmechanismen vorgeschlagen.

Die bisher identifizierten Use Cases umfassen:

- Link Becomes Available
- Ground Peer Publishes
- Airborne Peer Publishes
- Ground Peer Subscribes
- Airborne Peer Subscribes

- Ground Peer Unsubscribes
- Airborne Peer Unsubscribes
- Link Becomes Unavailable
- Airborne Peer Unsubscribes
- Subscription Manager Reconnects
- Subscription Times Out
- Notification Delivery Fails

Schließlich wurde der Einsatz von Filtern und ihre entsprechenden Formate (basierend auf XML) spezifiziert. Filter sind wichtig für die effiziente Weiterleitung von Nachrichten an interessierte Applikationen oder Nutzer (subscribers).

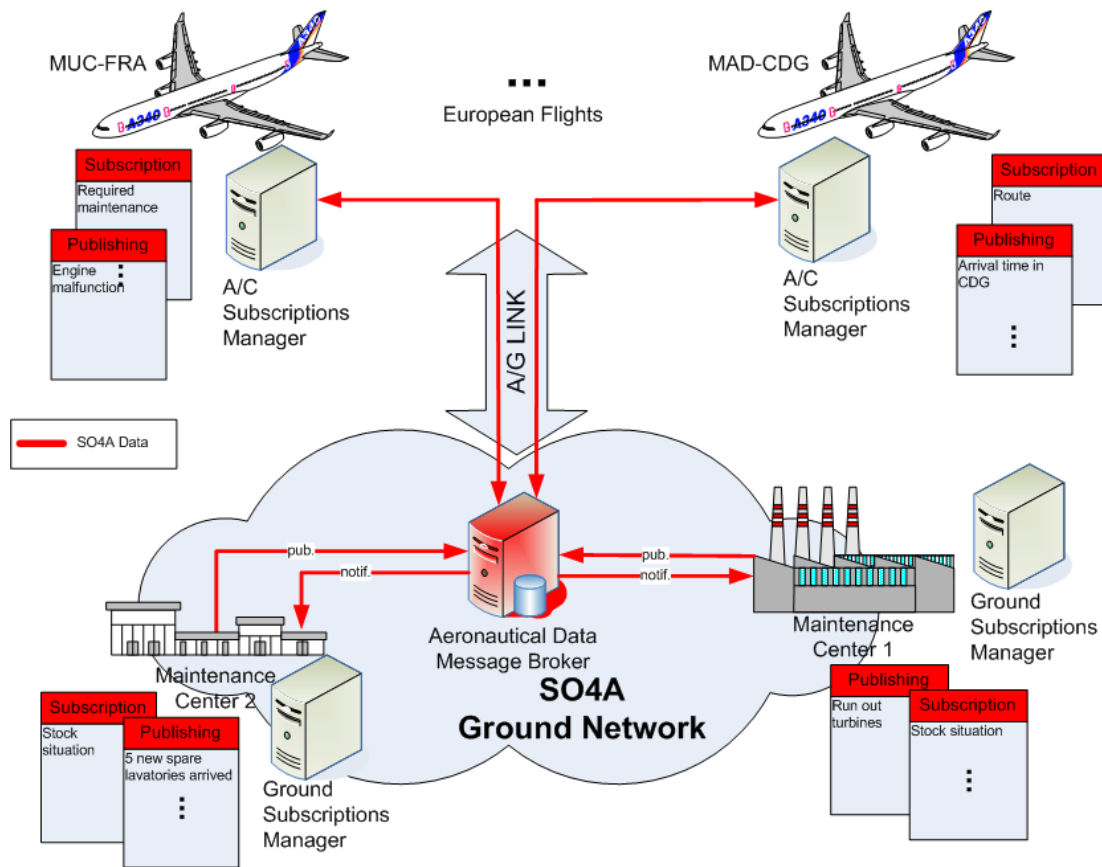


Abbildung 22: SO4A Kommunikations-Architektur

### 2.2.3 AP 141: Erstellung Funktionsmuster der Kommunikationsinfrastruktur

[Kurzfassung der geplanten Arbeiten laut Vorhabenbeschreibung:

Die Implementierung der in AP132 definierten Subsysteme erfolgt unter Führung von TGS in mit Unterstützung/Zuarbeit von Airbus und TU Ilmenau:

- Implementierung einer Dienste-Infrastruktur zur Dienstintegration
- Demonstration der Dienstintegration und deren Strategien
- Implementierung einer erweiterten und eigenständigen SOA-Labor-Testumgebung.]

Das System wurde nach Spezifikation aus AP 132 komplett implementiert und erfolgreich getestet.

Bei der Implementation der System-/Softwarekomponenten der SOA-Testumgebung werden C++ und XML-RPC verwendet. Alle folgenden Software-Komponenten, Software-Applikationen und Nachrichten wurden implementiert und getestet:

- Software-Komponenten: Airborne Broker, Airborne Peer, Ground Broker, Ground Peer
- Software Applikationen und Dienste: Link Controller, Klient oder Dienst (z.B. Meldungen über fehlerhafte Teile, Wetter-Dienst)
- Nachrichten: Broker Subscription, Notification, Publish, Unsubscription.
- 

Das Klassendiagramm der verschiedenen Komponenten ist in Abbildung 23 dargestellt. Abbildung 25 zeigt den Setup der Testumgebung der SO4A-Kommunikationsinfrastruktur, mit einem Ground Broker, einem Link Controller und mehreren Flugzeugen, jedes davon mit einem oder mehreren Airborne Peers, und einem Subscriptions Manager.

Der Betrieb von Klienten oder Diensten sowie vom Link Controller lief in einer ersten Implementierungsversion über eine GUI (Nutzeroberfläche) und erfordert die Interaktion von einem Benutzer. Das heißt, der Benutzer kann ein Ereignis über eine Web-Oberfläche eingeben (z.B. fehlerhaftes Teil x im Flugzeug y), und dadurch wird automatisch eine Notification generiert, der Ground Broker wird abhängig vom Link Status (geliefert vom Link Controller) informiert, und die Nachricht wird an die interessierten (subscribed) Ground Peers weitergeleitet. Die Weiterleitung von Nachrichten wird behandelt durch passende Filter im Ground Broker.

Im nächsten Schritt erfolgte die Automatisierung der Testumgebung mit dem Ziel, Testabläufe ohne manuelle Interaktion von Nutzern durchführen zu können. Dafür wurde die Implementierung der System-/Softwarekomponenten der SOA-Testumgebung erweitert, um eine Schnittstelle zu dem ML-Designer SO4A-Simulator der TU Ilmenau zu ermöglichen: der SO4A-Simulator generiert eine Datei (in einem vordefinierten XML-Format), die als Input für das SO4A Kommunikations-Testbed dient.

Diese Erweiterung umfasst nicht nur einer Software-Funktion, die diese Datei liest, sondern auch die Umsetzung von mehreren Blöcken für die Automatisierung der Testumgebung. Die endgültige Version des Funktionsmusters der Kommunikationsinfrastruktur erlaubt Tests nach manuellen Eingaben des Nutzers über eine GUI, aber auch automatisierte Tests.

Der Output eines automatisierten Tests ist wiederum auch eine XML-Datei mit der Zusammenfassung der Ereignisse, die während des Tests stattgefunden haben, womit man dem Datendurchsatz in einem Link und die Nachrichtenverzögerung berechnen kann.

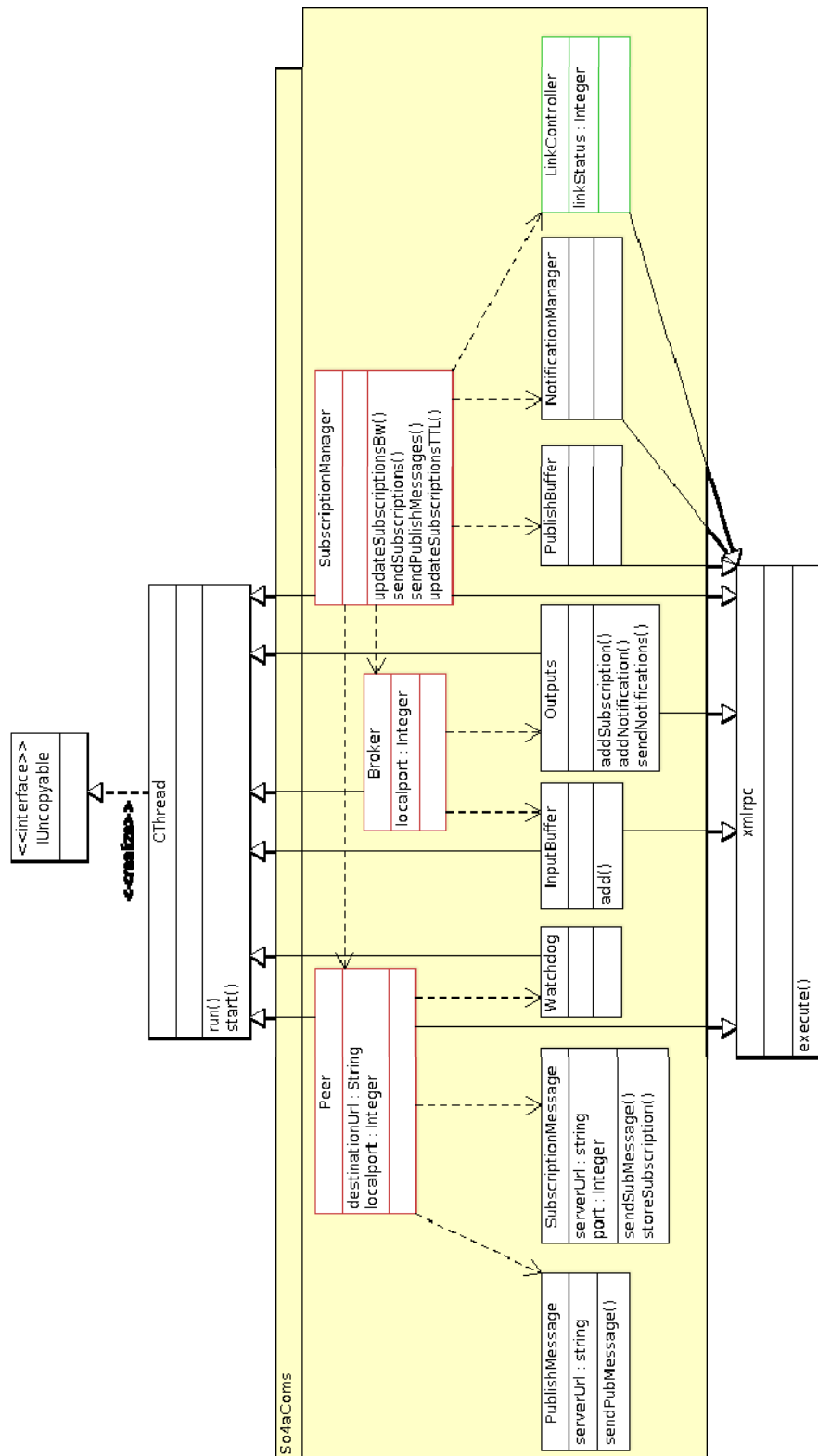


Abbildung 23: Klassendiagramm der SO4A Software-Implementierung



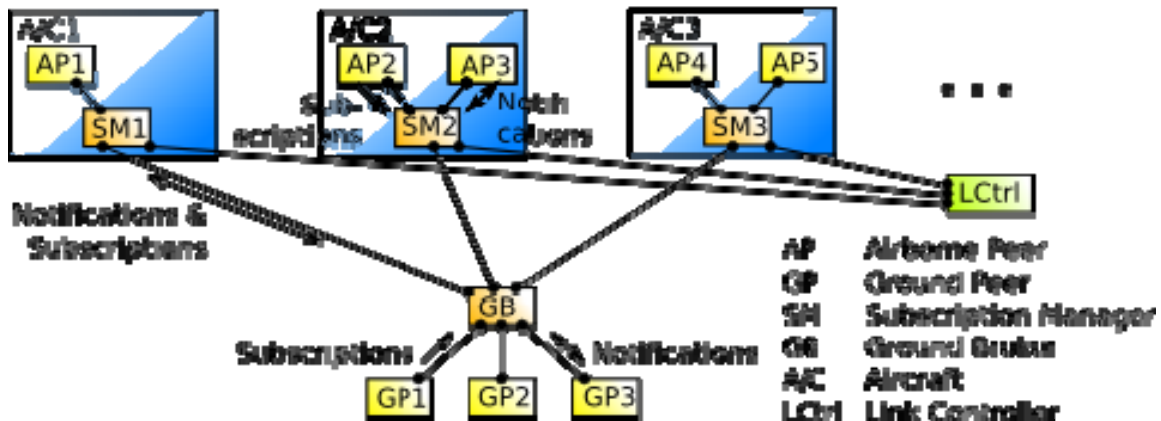


Abbildung 24: Setup der SO4ATestumgebung

Nach der Ausführung der Implementierung wurde der Fokus der Arbeiten in diesem AP auf die Tests gelegt. Zunächst wurde die Testumgebung in einer virtuellen Umgebung umgesetzt, damit Tests mit mehreren (emulierten) Flugzeugen möglich waren. Abbildung 25 erläutert die Zuordnung zwischen SO4A Software-Komponenten und die drei eingerichtete Virtuelle Maschinen (VM):

- VM1 (erste Virtuelle Maschine):  
 An jedem Port der VM1 hängt ein SM (Subscriptions Manager), der mit mehreren AP (Aircraft Peers) kommunizieren kann. Die APs informieren ihren SM über bestimmte Ereignisse. Die Sub-Gruppen von SM und APs entsprechen Flugzeugen. Zusätzlich kann jeder SM eine Verbindung mit dem GB (Ground Broker) erstellen, bzw. sich registrieren;
- VM2 (zweite Virtuelle Maschine):  
 Die VM2 enthält das GB, zuständig für das Weiterleiten von Nachrichten, das Pflegen von Registrierungen und von Filterregeln;
- VM3 (dritte Virtuelle Maschine):  
 An jedem Port der VM3 hängt ein GP (Ground Peer). Die GPs könne sich für bestimmte Ereignisse subscribieren. Nur dann werden sie die damit bezogene Information bekommen.

Die Eigenschaften der Kommunikationslink zwischen SMs und GB ist in der Software konfigurierbar. Folgende zwei Konfigurationen wurden für Tests im Rahmen dieses AP verwendet:

- ein idealer Link, ohne Verzögerung und mit einer gesamten Kapazität von 5 Mbps; diese Konfiguration erlaubte die Auswertung der Kommunikationsinfrastruktur praktisch ohne Limitierungen durch den Link;
- ein Satelliten-ähnlicher Link, mit 1.6 ms Verzögerung und einer Kapazität von 128 kbps pro Flugzeug.

Natürlich sind andere Konfigurationen möglich.



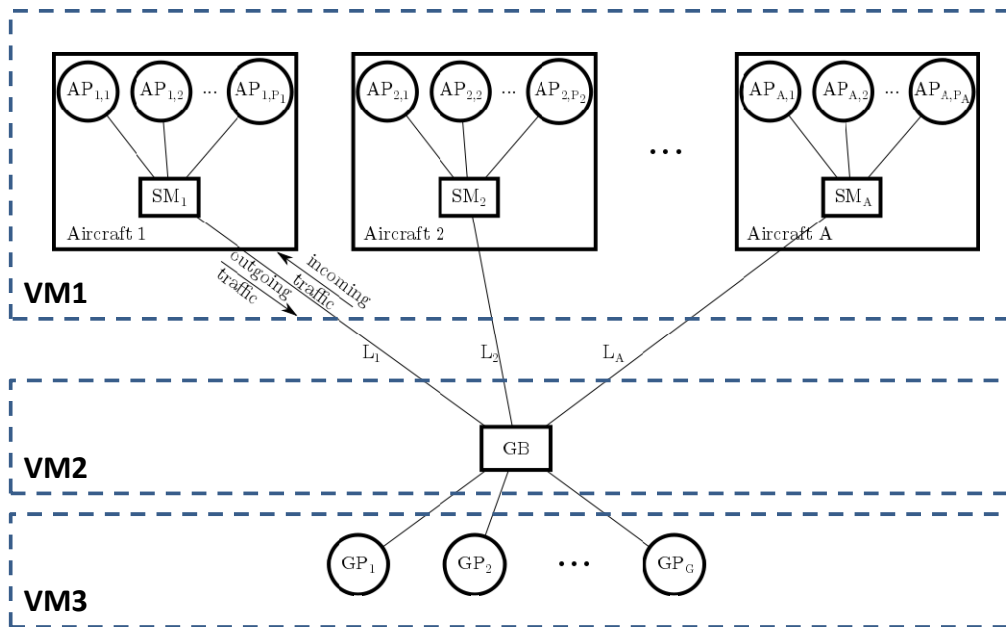


Abbildung 25: SO4A Software-Implementation in einer virtuellen Testumgebung

Die SOA-Testumgebung im TGS-Labor weist klare Schnittstellen zu ML-Designer-Modellen und der Simulationsumgebung von TU Ilmenau und Airbus auf; insbesondere werden Anforderungsdaten aus dem Flottenmodell der TU Ilmenau zur Ansteuerung des Kommunikations-Testbeds verwendet. Im Gegenzug werden vor allem relevante Parameter wie die Verzögerungszeiten von Nachrichten, die aus dem Testbed gewonnen wurden, in die übergeordnete ML-Designer-Modellierung eingespeist. Damit wird das Modell realistischer, und der Einfluss der Kommunikation auf dem Gesamtprozess kann simuliert werden.

Eines der wichtigsten Ergebnisse bei den Tests war die Feststellung der Abhängigkeit der Übertragungsverzögerung mit der Meldungs-Verzögerung, die in dem SM einstellbar ist. Außerdem bestätigte sich in den Untersuchungen quantitativ die Erwartung, dass mit recht niedrigem (erzeugten) Wartungs-Kommunikationsverkehr (Inputs aus der XML-Datei der TU Ilmenau) die resultierenden OPEX-Kosten einer solchen Kommunikationsinfrastruktur im sinnvollen Rahmen gehalten werden können. Konkret war der Kommunikationsverkehr in verfügbaren Beispiel (mit realistischen Wartungs-Ereignissen erzeugt) nur ca. 98 kbyte pro Stunde.

Daher zeigt sich die ausgewählte Publish/Subscribe SO4A/SWIM-Kommunikationsinfrastruktur als sehr geeignet für Wartungsdienste.

Die Konfigurierbarkeit des implementierten Kommunikationstestbeds erlaubt die Untersuchung der Verwendung dieser Infrastruktur bei anderen Diensten.

## 2.3 Wichtigste Positionen des zahlenmäßigen Nachweises

Die vorhabenbezogene Mittelverwendung erfolgte weitgehend gemäß Mittelplanung zielgerichtet für die Durchführung der in den vorangegangenen Abschnitten detaillierten Aktivitäten.

Das Teilvorhaben wurde von Kommunikationsingenieuren, Informatikern und Wissenschaftlern bei TriaGnoSys ausgeführt; für unterstützende Arbeiten werden auch Werkstudenten/Diplomanden herangezogen. Je nach Arbeitspaket und Zeitpunkt im Projekt waren üblicherweise mehrere Personen an der Durchführung der Arbeiten beteiligt. Über die gesamte Projektlaufzeit standen jeweils ein erfahrener Wissenschaftler/Teilprojektleiter sowie ein erfahrener Kommunikationsingenieur und ein Informatiker/Software-Architekt mit einem wesentlichen Anteil ihrer Arbeitszeit für das Projekt zur Verfügung.

Der weitaus größte Teil der für die Durchführung des Vorhabens eingesetzten Mittel sind personenbezogene Kosten. Die TGS-Ressourcen verteilen sich zu ca.75% auf SWAN-TGS und 25% auf SO4A-TGS.

Die Kosten für das Vorhaben sowie die bewilligte Zuwendung (bei 50% Eigenanteil des Zuwendungsempfängers) umfassten folgende Positionen, die weitgehend nach Plan wie folgt verwendet wurden:

### 1. Personalkosten – 99%

Der dominierende Kostenanteil waren die Personalkosten, umfassend die direkten und indirekten Kosten für Projektleiter, Wissenschaftler, Ingenieure, Programmierer und Werkstudenten/Diplomanden. Der Gesamtumfang der geleisteten Arbeiten betrug ca. 88 Personenmonate.

### 2. Vorhabensspezifische Abschreibungen – <1%

Für den Laboraufbau der Kommunikationsinfrastruktur vor allem im Teilprojekt SWAN wurden einige Industrie-PCs gekauft, sowie für die hardwarenahe Entwicklung der WSN-Software eine Entwicklungsumgebung.

### 3. Reisekosten – < 1%

Die Reisekosten für regelmäßige Projekt- und Arbeitstreffen waren aufgrund der regionalen Nähe der Verbundpartner sehr gering.

## 2.4 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die Risiken des Forschungsvorhabens @MOST liegen in einem hohen Innovationsgrad und in der Komplexität der zu lösenden Aufgaben.

Die im Teilprojekt @MOST-SWAN adressierten selbstorganisierenden Sensornetzwerke für den Dauerbetrieb sind eine aufkommende Technologie, deren Leistungsfähigkeit im praktischen Einsatz immer noch nicht völlig erforscht ist. Dazu kommt erschwerend hinzu, dass der Einsatz in der Flugzeugumgebung sehr hohe Anforderungen bezüglich Kommunikationssicherheit und Zuverlässigkeit der Hardware und Software in widrigen Umgebungsbedingungen stellt. Die damit verbundenen Unwägbarkeiten bezüglich Machbarkeit und nicht zuletzt Zulassbarkeit stellen für das Vorhaben eine große Herausforderung dar und erfordern deshalb umfangreiche Arbeiten von grundlegenden Aspekten bis hin zur Implementierung von Teil- und Systemlösungsansätzen auf angemessenem TRL, welches nur im Verbund geeigneter Partner mit entsprechendem Ressourcenaufwand zu leisten ist.

Die im Bereich des Teilprojekts @MOST-SO4A entwickelte service-orientierte Architektur ist von hoher Komplexität, da die Bandbreite der zu berücksichtigenden Prozesse sich über viele Größenordnungen bezüglich der zu wartenden Funktionseinheiten erstreckt: Umfassende Kosteneinsparungen bei der Wartung sind nur dann zu erwarten, wenn, angefangen von speziellen Flugzeugkomponenten bis hinauf zum Flottenmanagement, möglichst alle relevanten Wartungsprozesse erfasst und analysiert werden. Diese Aufgabe erfordert ein wohl koordiniertes Zusammenspiel eines interdisziplinären Forschungskonsortiums; insbesondere hängt der zielgerichtete Nutzen der von TriaGnoSys vornehmlich bearbeiteten Kommunikationsinfrastruktur maßgeblich vom Erfolg der gesamten Prozesskette ab.

Die Ausgangslage für mögliche Ansätze von Kommunikationslösungen in beiden Teilprojekten ist somit sehr breit, so dass echte Grundlagenarbeit zu leisten ist in einem umfangreichen Prozess (mit mehrstufigen Iterationen zwischen Partnern) von der Ermittlung der Anforderungen, über Definition der möglichen Systemarchitekturen und Design/Implementierung von Subsystemlösungen, der nachfolgenden Integration in Labormuster und Testbeds, bis hin zur Demonstration im Mockup.

Der erforderliche Ressourcenaufwand für die Umsetzung des Teilvorhabens war deshalb erheblich, im geleisteten Umfang aber u.a. auch möglich aufgrund der eigenen Vorarbeiten zu relevanten Teilaspekten aeronautischer Kommunikation. Allerdings wurde darauf aufbauend weitgehend Neuland in Bezug auf das Gesamtvorhaben beschritten.

## 2.5 Voraussichtlicher Nutzen und Verwertung der Ergebnisse

Der allgemeine Verwertungsplan des Verbundvorhabens @MOST-G trifft in wesentlichen Punkten auch auf TriaGnoSys zu und wird durch die folgenden Ausführungen um TGS-spezifische Anteile entsprechend erweitert.

Der Verwertungsplan für das Teilvorhaben von TriaGnoSys wurde im Zuge der Umstellung des Verbundvorhabens grundlegend neu gefasst, da sich die strategische Bedeutung mit dem Wechsel vom Status als Airbus-Unterauftragnehmer in den Status eines gleichberechtigten Verbundpartners neu darstellte. Gegen Ende des Projekts verdichteten sich die Anzeichen für die Berechtigung der formulierten Erwartungen und Pläne durch verstärkte Diskussionen mit verschiedenen, auch neuen Kooperationspartnern und Technologieherstellern. Konsequenterweise wurde mit NELA-HyWAYS ein entsprechend zugeschnittenes Nachfolgeprojekt begonnen, welches vor allem einen höheren Reifegrad (TRL) der entwickelten prototypischen Lösungen vor allem im WSN-Bereich anstrebt.

Die drahtlose Sensornetzwerk-Technologie, insbesondere in Kombination mit Server- und Anwendungssoftware, ist für TriaGnoSys als spezialisiertem Subsystem-Lieferanten und Systemintegrator für drahtlose Kommunikationslösungen von strategischer Bedeutung:

- Erweiterung der bisherigen aeronautischen Mobilkommunikations- und drahtlosen Internetlösungen für Passagiere durch sensorbasierte Datenanwendungen für Maintenance etc.;
- Beherrschung/Nutzung der Technologie und Software auch in anderen relevanten Bereichen, z.B. Umweltmonitoring; Sicherheitsnetze, verteilte intelligente Energiesysteme.

Längerfristig plant TriaGnoSys eine aeronautisch zertifizierte WSN/WDC-Lösung auf den Markt zu bringen.

Die mittel- bis langfristige Planung von TriaGnoSys sieht vor, sowohl im Bereich Kommunikations-, Server- und Anwendungssoftware für drahtlose Sensor-Kabinennetze (SWAN-TGS) wie auch im Bereich SoA-basierte Anwendungen für verschiedenste aeronautische Anwendungen (SO4A-TGS) zielgerichtet Nischenprodukte zu entwickeln.

Aus heutiger Sicht sind die Arbeiten im @MOST-G-Konsortium international führend, und im Verbund mit den großen Partnern und jeweiligen Marktführern erscheinen die Ambitionen des Projekts nach Lösung der wesentlichen technischen Herausforderungen auch ökonomisch realisierbar.

Für die positive Einschätzung der wirtschaftlichen Erfolgsaussichten sind für TriaGnoSys vor allem auch folgende Aspekte bedeutsam:

- Die drahtlose Sensornetzwerk-Technologie kann komplementär zu existierenden erfolgreichen Server- und Anwendungssoftware-Produkten von TriaGnoSys entwickelt und eingeführt werden, nutzt also einen Marktvorsprung und existierende Kundenbeziehungen.
- Sensorbasierte Datenanwendungen z.B. für Maintenance stellen aus Sicht der Kommunikationsinfrastruktur und –software eine logische Erweiterung der bisherigen aeronautischen Mobilkommunikations- und drahtlosen Internetlösungen für Passagiere dar.
- Synergien für WSNs ergeben sich auch mit laufenden Arbeiten zur Cockpit-Kommunikation.
- Beherrschung/Nutzung der Technologie und Software auch in anderen relevanten Bereichen, z.B. Umweltmonitoring; Sicherheitsnetze, verteilte intelligente Energiesysteme.
- Im Bereich Service-Oriented Architecture (SoA) ergeben sich für TriaGnoSys weitreichende Synergien mit laufenden und künftigen Projekten und Produktentwicklungen in den Bereichen (i) Neuartige Anwendungen für zukünftige Flugverkehrskontrolle (SoA-basierte SWIM-Anwendungen – System Wide Information Management), (ii) dienstorientierte Wetter- und Meteo-Applikationen für Flugzeuge mit/ohne Integration von Wettersensoren und (iii) verteilte intelligente und nachhaltigkeitsorientierte Logistik- und Produktionsprozesse.
- Insgesamt sind praktisch alle in @MOST von TGS entwickelten Komponenten auch für das (von @MOST insgesamt zunächst nicht adressierte) zusätzliche Marktsegment der kleinen und Geschäftsflugzeuge interessant.

Mit den aufgeführten Punkten wird offensichtlich, dass aus Sicht von TriaGnoSys genügend kommerzielles Potenzial bzw. zumindest wirtschaftliche Risikominimierung gegeben scheint.

Die enge Einbindung und Verknüpfung der verschiedenen für das Forschungsvorhaben @MOST-G relevanten Disziplinen aus den Bereichen der IT-Dienstleistung sowie der Flugzeugkomponentenzulieferindustrie eröffnet immense Möglichkeiten zur frühzeitigen Berücksichtigung von ganzheitlichen dienstorientierten Wartungsansätzen für künftige Airbus Großflugzeuge. Durch die Führung des Projektes seitens Airbus ist gewährleistet, dass diese neuen Ansätze bereits im Vorfeld eines neuen Flugzeugprogramms Berücksichtigung finden. Zudem wurden die Projektpartner sorgfältig ausgewählt, um die Überführung der erarbeiteten Technologien in ausgereifte Produkte und Dienstleistungen sicherzustellen. Es besteht ein großes Potential, dass sich durch die Einführung der für die in @MOST-G untersuchten dienstorientierten Ansätze zum Flottenbetrieb völlig neue Geschäftsmodelle für etablierte und neu entstehende Unternehmen ergeben.

Die Ergebnisse werden gemeinsam mit den Verbundpartnern entwickelt und münden in einer Verifikationsplattform, an der TriaGnoSys starke Implementierungsanteile hat. Es besteht damit die Aussicht, dass relevante TGS-Anteile auch nach Projektende in den Airbus-Flugzeugprogrammen verwertet werden können. Dadurch wird das Know-how und die wirtschaftlich-technische Situation der Luftfahrtindustrie und des Luftverkehrs verbessert und vor allem die Wettbewerbs- und Innovationsfähigkeit sowie die Kompetenz auf nationaler Ebene gestärkt.

Dieses Forschungsprojekt trägt damit einerseits zur Sicherung der bestehenden Arbeitsplätze von TriaGnoSys bei, andererseits wird ein zukünftiger kommerzieller Erfolg neue Arbeitsplätze in diesem Geschäftsfeld schaffen, welches zudem die heutige führende Position bei Kabinenkommunikation auf dem Weltmarkt sinnvoll ergänzt und stärkt.

## **2.6 Während des Vorhabens bekannt gewordener Fortschritt Dritter**

Auf dem speziellen Gebiet der Teilprojekte SWAN und sind dem Zuwendungsempfänger während der Projektlaufzeit keine aktuellen Ergebnisse bzw. Fortschritt Dritter bekannt geworden, die für die Durchführung des Vorhabens spezifisch relevant gewesen wären.

Zu erwähnen ist für das Teilprojekt SWAN allerdings im positiven Sinn ein verwandtes Vorhaben von Airbus, nämlich das Projekt *CabWiSe*, in dem die Firma Funkwerk Avionics, an einer eigenen neuen Hardware/Software-Plattform für künftige kommerzielle WDCs arbeitet. TriaGnoSys ist dem Vorhaben CabWiSe kostenneutral mit dem Ziel beigetreten, relevante Informationen und Spezifikationen zu dieser Plattform nach Möglichkeit rechtzeitig in den verbleibenden SWAN-Design- und Implementierungs-Arbeiten berücksichtigen zu können. Dies geschah und geschieht nun vor allem durch regelmäßige Treffen und beiderseitigen Informationsaustausch der Firmen Funkwerk Avionics und TriaGnoSys.

## **2.7 Veröffentlichungen der Ergebnisse**

In Absprache mit dem Verbundführer Airbus, vor allem aus Gründen der Wahrung des erarbeiteten Innovationsvorsprungs in der wichtigen Zukunftstechnologie Sensornetze, erfolgten keine allgemeinen Veröffentlichungen außer dem vorliegenden Bericht.

Dies hängt insbesondere damit zusammen, dass die Teilarbeiten von TriaGnoSys sehr detailliert vor allem ins Software-Design sowie in die Implementierung gehen. In diesem Bereich sind nützliche Veröffentlichungen ohne die Offenlegung von signifikanten IPR/Knowhow-relevanten Anteilen kaum möglich.

Im Zusammenhang mit den laufenden Nachfolgeprojekten ist in den kommenden Jahren auch an allgemeinere wissenschaftliche Veröffentlichungen gedacht, vor allem wenn mit Messergebnissen aus Labormusteraufbauten entsprechende gut darstellbare Ergebnisse vorliegen.

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
AOA	Autonomous Operations Area
AP	Arbeitspaket, Access Point
BMWi	Bundesministeriums für Wirtschaft und Technologie
EADS IW	European Aeronautic Defence and Space Company N.V. Innovation Works
FTI	Flight Test Installation
GB	Ground Broker
GP	Ground Peer
HMI	Human Machine Interface
ICD	Interface Control Document
IEEE	The Institute of Electrical and Electronics Engineers
IT	Information Technology
LAMT	Location Aware Maintenance Tool
@MOST	Aircraft Total Maintenance Operations, Solutions & Technologies Program
PC	Personal Computer
PDA	Personal Digital Assistant
RPC	Remote Procedure Call
RSSI	Receive Signal Strength Indicator
SESAR	Single European Sky Aeronautics Research
SIL	System Integration Laboratory
SM	Subscription Manager
SO4A	Service Oriented Architecture Application for Airlines & Aircraft
SOA	Service Oriented Architecture
SQL	Structured Query Language
SWAN	Sensor Wireless Application Network
SWIM	System Wide Information Management
TGS	TriaGnoSys GmbH
TRL	Technology Readiness Level
TU	Technical University
TUHH	Technical University Hamburg-Harburg
USB	Universal Serial Bus
VM	Virtual Machine

Abkürzung	Bedeutung
WDC	Wireless Data Concentrator
WSN	Wireless Sensor Network, Wireless Sensor Node
XML	eXtensible Markup Language

## Literaturverzeichnis

- [1] Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, E. Cayirci: Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.
- [2] Kansal, A. and Srivastava, M. B. 2003. An environmental energy harvesting framework for sensor networks. In *Proc. int. Symp. on Low Power Electronics and Design, ISLPED '03*, 481-486, 2003.
- [3] Demirkol, I., Ersoy, C. Alagöz, F.: MAC Protocols for Wireless Sensor Networks: A Survey, *IEEE Communications Magazine*, 115–121, 2006.
- [4] Maroti, M., Kusy, B., Simon, G., Lédeczi, A.: The flooding time synchronization protocol. In: *Proc. 2nd Int. ACM Conf. on Embedded Networked Sensor Systems (SenSys 04)*, 39-49, 2004.
- [5] The TinyOS web site: <http://www.tinyos.net/>.
- [6] Madden, S., Franklin, M., Hellerstein, J., Hong, W.: TinyDB: An Acquisitional Query Processing System for Sensor Networks. *ACM TODS*, 2005.
- [7] Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks, *Ad Hoc Networks* 3, 325–349, 2005.
- [8] Witt, M.: Matthias Witt: Robust and Low-Communication Geographic Routing for Wireless Ad-hoc Networks, Dissertation, TU Hamburg-Harburg, 2008.
- [9] Turau, V., Weyer, Ch.: Fault Tolerance in Wireless Sensor Networks through Self-Stabilization, *Int. Journal of Communication Networks and Distributed Systems*, 2008.
- [10] Lu, J.-L., Valois, F., Barhel, D.: Low-Energy Self-organization Scheme for Wireless Ad Hoc Sensor Network. In *4th Annual Conf. on Wireless On demand Network Systems and Services (WONS)*, 2007.
- [11] Liu, D., Ning, P.: *Security for Wireless Sensor Networks*, Springer Verlag, 2007.
- [12] Ranging with IEEE 802.15.4 Narrow-Band PHY, IEEE Dokument 802.15-09-0613-00-004f, 14-09-2009, veröffentlicht unter <https://mentor.ieee.org/802.15/dcn/09/15-09-0613-00-004f-rangingwith-ieee-802-15-4-narrow-band-phy.ppt>



## Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel  TriaGnoSys Schlussbericht @MOST-SWAN/SO4A Vorhaben @MOST-G – Aircraft Total Maintenance Operations, Solution & Technologies Program – Germany Teilvorhaben Wireless Sensor Network Infrastructure and Software (SWAN-TGS) & SoA Air-Ground Communications Infrastructure (SO4A-TGS)	
4. Autor(en) [Name(n), Vorname(n)] Riera, Nuria Dupont, Emeric Kühndel, Frank Valcarce, Alvaro Werner, Markus	5. Abschlussdatum des Vorhabens 30.06.2012
	6. Veröffentlichungsdatum 15.06.2013
	7. Form der Publikation Broschüre
8. Durchführende Institution(en) (Name, Adresse)  TriaGnoSys GmbH Aargelsrieder Feld 22 82234 Weßling	9. Ber. Nr. Durchführende Institution -
	10. Förderkennzeichen 20K0801F
	11. Seitenzahl 47
12. Fördernde Institution (Name, Adresse)  Bundesministerium für Wirtschaft und Technologie (BMWi) 53107 Bonn	13. Literaturangaben 12
	14. Tabellen -
	15. Abbildungen 25
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum)	
18. Kurzfassung 1. Derzeitiger Stand von Wissenschaft und Technik: Wartungszyklen für Flugzeuge sind heute oft noch unnötig lange aufgrund von (i) unzureichender/später Datenlage über die durchzuführenden Wartungsarbeiten; (ii) ineffizienter Kommunikation der an der Wartungskette beteiligten Partner. 2. Begründung/Zielsetzung der Untersuchung: (i) Einsatz von drahtlosen Sensornetzen an Bord zur effizienteren Gewinnung von relevanten Daten. (ii) Aufbau einer dienstorientierten Ende-zu-Ende-Kommunikationsinfrastruktur zwischen Flugzeug und Boden auch während des Fluges. 3. Methode: Aus Nutzer- und Systemanforderungen wurde eine umfassende Architektur für eine flexible Kommunikationsinfrastruktur abgeleitet. Diese wurde gemäß einem systematischen Hardware/Softwaredesign implementiert, aufgebaut, im Labor und Mockup getestet und schließlich Nutzern demonstriert. 4. Ergebnis: Für (i) und (ii) jeweils ein verifiziertes Konzept, implementiertes integriertes Testbed mit Laborprototypen, und Messergebnisse. 5. Schlussfolgerung/Anwendungsmöglichkeiten: Beide vorgenannten Teilsysteme zeigen großes Potenzial für die mittelfristige Umsetzung in künftiger Flugzeugtechnologie sowie operationellen Prozessen, und insbesondere die Integration beider Teilsysteme verspricht weitere Synergien in den Wartungs- und Betriebsabläufen.	
19. Schlagwörter Drahtlose Sensornetze, Kommunikationsinfrastruktur, dienstorientierte Architektur, Flugzeuge	
20. Verlag	21. Preis

## Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) report
3. title  TriaGnoSys Schlussbericht @MOST-SWAN/SO4A Vorhaben @MOST-G – Aircraft Total Maintenance Operations, Solution & Technologies Program – Germany Teilvorhaben Wireless Sensor Network Infrastructure and Software (SWAN-TGS) & SoA Air-Ground Communications Infrastructure (SO4A-TGS)	
4. author(s) (family name, first name(s)) Riera, Nuria Dupont, Emeric Kühndel, Frank Valcarce, Alvaro Werner, Markus	5. end of project 30.06.2012
	6. publication date 15.06.2013
	7. form of publication brochure
8. performing organization(s) (name, address)  TriaGnoSys GmbH Aargelsrieder Feld 22 82234 Weßling	9. originator's report no. -
	10. reference no. 20K0801F
	11. no. of pages 47
12. sponsoring agency (name, address)  Bundesministerium für Wirtschaft und Technologie (BMWi) 53107 Bonn	13. no. of references 12
	14. no. of tables
	15. no. of figures 25
16. supplementary notes	
17. presented at (title, place, date)	
18. abstract 1. State-of-the-art: Aircraft maintenance cycles often suffer from unnecessary delays due to (i) insufficient data concerning required maintenance operations; (ii) inefficient communication along the maintenance supply chain. 2. Motivation/objectives of the study: (i) Use of on-board wireless sensor networks for more efficient gathering of relevant maintenance data; (ii) Implementation of a service-oriented end-to-end communication infrastructure between aircraft and ground also during flight 3. Method: Based on user and system requirements, a comprehensive architecture for a flexible communication infrastructure was derived. According to a systematic hardware/software design, it was then implemented, integrated, tested and verified both in laboratory and in mockup, and demonstrated to users. 4. Results: For both subsystems, (i) and (ii), respectively, a verified concept, implemented integrated test-bed with lab prototypes, and measurement results. 5. Conclusion/application possibility: Both sub-systems show significant potential for medium-term implementation and use in future aircraft technology and related operational processes, and especially the integration of both sub-systems promises further synergies within the maintenance and operational procedures.	
19. keywords Wireless sensor networks, communications infrastructure, service-oriented architecture, aircraft	
20. publisher	21. price