ARWINET GmbH
Ziegelei 9, 72336 Balingen

https://www.arwinet.com/
projekte@arwinet.com

QUALIFIKATIONSPROFIL #10309

ALLGEMEINE DATEN

Geburtsjahr: 1972

Ausbildung: Abitur
Diplom, Informatik, (TU, Kaiserslautern)

Fremdsprachen: Englisch, Französisch

Spezialgebiete: Kubernetes

KENNTNISSE

Tools

Active Directory
Apache
Application
Case
CATIA
CVS
Eclipse
Exchange
Framework
GUI
Innovator
ITIL
J2EE
JMS
LDAP
Lotus Notes
make
MS Exchange
MS Outlook
MS-Exchange
MS-Office
MS-Visual Studio
NetBeans
OSGI
RACF
SAS
sendmail
Together
Turbine
UML
VMWare
.NET
ADS
ANT
ASP
ASP.NET
Flash
GEnie
IDES
Image
Intellij IDEA
IPC
Jackson
JBOSS
Lex
MS-Visio
ODBC
Oracle Application Server
OWL
PGP
SPSS
SQS
TesserAct
Tivoli
Toolbook
Total
Transform
Visio
Weblogic
WebSphere
YACC

Tätigkeiten

Administration
Analyse
Beratung
Design
Dokumentation
KI
Konzeption
Optimierung
Support
Vertrieb

Sprachen

Ajax	Basic
C	C#
C++	Cobol
Delphi	ETL
Fortran	Java
JavaScript	Natural
Perl	PHP
PL/I	PL-SQL
Python	SAL
Smalltalk	SQL
ABAP	Atlas
Clips	Delta
FOCUS	HTML
Nomad	Pascal
SPL	Spring
TAL	XML

Detaillierte Komponenten

AM	BI
FS-BA	MDM
PDM	PM
BW	CO
FI	LO
PP	

Datenbanken

Approach	IBM
Microsoft	Object Store
Oracle	Progress
Sybase	DMS
ISAM	JDBC
mySQL	

DC/Netzwerke

ATM	DDS
Gateway	HBCI
Hub	Internet
Intranet	OpenSSL
SSL	VPN
Asynchronous	CISCO Router
DNS	DSL
Firewall	Gateways
HTTP	RFC
Router	Samba
Sockets	Switches

Finance

Business Intelligence	Excel
Konsolidierung	Management
Projektleiter	Reporting
Testing	Wertpapiere
Einkauf	

CAD Systeme

CATIA V5	sonstige

Hardware

Digital	HP
PC	Scanner

Siemens	Spark
Teradata	Bus
FileNet	NeXT
SUN	Switching

Tools, Methoden

Docker	Go
Kubernetes	Rational
RUP	V-Modell
Ansible	HIL
MCS	

Mikrocontroller / DSP

Intel

Betriebssysteme

Linux	Unix
Windows	CMS

Pharma

Marktforschung

Bereich

Maschinenbau	Medizintechnik
Automotive	

SAP Keywords

Netweaver	Portal
SOA	Workflow
Migrations	

Funktionen

Packaging	Planung
Projektmanagement	Applikation
Diagnose	Qualitätssicherung

Fachgebiete

Radiologie

SoftSkills

Einfühlungsvermögen	Networking

BERUFS- UND PROJEKTERFAHRUNGEN

3/2020 - 9/2020

IT-Architekt, Product Owner, Produktmanager, Coach (Freiberuflich)

Konzeption/Entwicklung eines eigenen Intelligent Automation Tools, strategisch-gesteuert mit neuen KI-Techniken wie z.B. Planung, probabilistischen Modellen, etc.: Intelligentes RPA (Robotic Process Automation), Computer Vision/OCR (Optical Character Recognition), ICR (Intelligent Content Recognition), IDP (Intelligent Document Processing) sowie natürlicher Sprachverarbeitung (NLP, Natural Language Processing).
Zweck: Automatisierung von Tätigkeiten:
Für Verwaltungstätigkeiten: Infos lesen, bearbeiten, extrahieren und anderweitig in IT-Systeme eingeben: In Dateien, APIs, (Web-)Formulare, UIs von Apps, etc.)
Für agile Prozesse: Möglichst viel automatisieren und Situationen für die Anwendung von mittlerweile über 1000 agilen Praktiken erkennen und empfehlen, insbesondere im Risiko- und Konflikt-Management, im Ergreifen von Chancen, Erkennung von Entscheidungspunkten, etc.
Für eLearning: Automatisiertes Vorschlagen von Tipps, Korrekturen, Übertragen von Daten, psychologische Elemente zur Motivation einbringen, etc. u.A. durch Injektion von JavaScript (z.B. als Avatar), das analysiert, Tipps gibt, Lernfragen stellt, etc.
Für Marktanalyse (Kunden, Lieferanten, Konkurrenten, Finanz- und Personalmarkt): Web-Mining, Web-Scraping, Informationsaufbereitung.

Erstellung einer Gesamt-Architektur über die Komponenten und wesentlichen Prozesse eines komplett neuen intelligent (Robotic Processs) Automation Tools zum Deployment sowohl in Data Center oder der Cloud über die Containierisierung mit Docker und Kubernetes/K8s mit Microservices, Web-APIs oder Serverless. Diese spezielle Gesamt-Architektur diente u.A. zur Diskussion mit externen Partnern / Startups über mögliche Kooperationen / Integrationen.
Management des Gesamt-Projekts und der Unterprojekte: Projektplan, Investitionsplan, Finanzplan, Risiko-Management erstellt; Business Case Berechnung, Unterauftrag-Vergaben nach Offshore/Nearshore.
Data- und ML-Workflow-Konzept erstellt mit Knative Eventing und Kubeflow.
Partnerschaftsgespräche mit führenden und innovativen RPA- und Automationsfirmen sowie Influencern: UI Path, Automation Anywhere, Blueprism, Jacada, Kofax, Pega, Progress Corticon, Softomotive (Robin), NICE, WorkFusion, Nividous, Antworks, Thoughtonomy, Edgeverve AssistEdge, Kryon/VirtualAI, Another Monday.
Kernarchitektur erstellt basierend auf non-linear Planning Komponenten erweitert um probabilistische Regeln mit MIT's Gen.jl + Julia MLJ.jl, Pyro, ProbLog und In-Memory- und Tiered-Memory- Architektur mit Apache Pulsar, Spark, Alluxio and Redis/PostgreSQL/Aerospike.
Verwendete/getestete/adaptierte non-linear Planning and Control Libraries: Control Toolbox, AIKIDO, ROS Navigation2+ROS Behavior-Tree, Open Motion Planning Library (OMPL).
MVP-Konzeption (Minimal Viable Product nach Lean Startup) und Implementierung mit Fokus auf web-basierte Techniken: Fokus auf TypeScript-/JavaScript-basierte Webseiten-Analyse, Extraktion und das Formular-Ausfüllen mit diesen Tools: CasperJS, PhantomJS, Slimer.js sowie GreaseMonkey/Tampermonkey (JS Code Injection), Selenium.
Datenspeicherung und Suche mit Solr; Datenextraktion / Web-Scraping u.A. mit Apache Tika / Gora / Nutch.
Optische Zeichenerkennung (Optical Character Recognition) integriert mit STN-OCR und neuen Computer Vision Techniken/Algorithmen.
React-basiertes UI erstellt für deklarative visuelle Automations-Spezifikationen (visuelle Verbindung von Schritten, Data Mapping) oder Implementierung in Robin RPA oder JavaScript.
Erstellung eines React-basierten UI für verteiltes Brainstorming und eLearning, in das Teilnehmer verteilt Rich-Texts und Concept Graphs/MindMaps erstellen können, um Themen zu diskutieren. Erweiterung um Auto-Korrektur, Kontext-basierte Vorschläge, Nachschlagen in Wikis/Wissensbasen, Unternehmens-Apps, Suche in E-Mails und sonstiger Smart Input, etc.
Injecten eines Analyse- und Avatar-JavaScript-Codes in eLearning Tools wie Moodle, ILIAS, WebUntis, etc. um Tipps zu geben, Kontrollfragen zu stellen, das Durchdenken der Inhalte zu fördern und diese zu wiederholen, z.B. nach dem Dreiske-Ansatz, der bei ca. 80% der Kinder ca. 70% verbesserte Denkfähigkeiten höherer Ordnung bringt: Inferentielles Denken, kritisches Denken und Metakognition – insbesondere durch Fragen nach den Zusammenhängen, Hintergründen, emotionalen/psychologischen Aspekten wie Einfühlungsvermögen und Konflikt-Verstehen & Konfliktlösung; weiterhin wird das Vergessen in den Sommerferien adressiert: Alles zunehmend automatisiert mit KI- und NLP-Techniken.
Erstellung eines React-basierten UI für die Analyse und Verbesserung von agilen Entwicklungsprozessen (AI-driven Agile) durch Analyse von Wikis, Dokumenten, E-Mail-Verkehr und auf dieser Basis Gabe von Empfehlungen. Hierfür wurden über 1000 Agile Best Practices aus dem Internet extrahiert und in eine Wissens-Datenbank aufbereitet, z.B. mindsettlers.com, myagile5.com, sociocracy30.org, holacracy.org, management30.com, Disciplined Agile Delivery (DAD) Bücher, Design Thinking Bücher wie (Das große Handbuch Innovation: 555 Methoden und Instrumente; 77 Tools für Design Thinker; Das Design Thinking Toolbook), etc.
Vorschlag/Konzeption und Integration von Data Science und KI-Algorithmen (mit SQL, ODBC, JDBC, etc.), insbesondere auch aus NLP (Google BERT und Nachfolger) und Computer Vision.
Akquise und (technische) Koordination von Offshore-Entwicklungspartnern.
Erstellung von Online-Bots in JavaScript zur Markierung, Extraktion, Speicherung und Weiterverwendung von Informationen, etwa für Social Media Marketing.
Zielgerichtete Analyse potentieller Kunden in verschiedenen Quellen (Websites, Datenbanken, Patente, Bewertungsportale, Jobportale, etc.) und strategisch bzgl. spezieller Fragestellungen (sortiert nach Prognose-Fähigkeit für positive Outcomes) zwecks möglichst detaillierter Kunden-Segmentierung und zielgenauer personalisierter Ansprache in E-Mails, Briefen, Telefonaten und in Chatbots; semantisches Content Marketing (Kunden-Push).
Systematische Konkurrenz- und Marktanalyse (Kunden, Lieferanten, Konkurrenten, Finanz- und Personalmarkt): Web-Mining, Web-Scraping, Informationsaufbereitung in relationale Strukturen, Triggern von regelbasierten und semantischen Alerts: Erkennen, Einschätzen und Reporten von Bedrohungen, Risiken, Chancen, wichtigen Events, Fördermitteln, etc.
Zielgerichtetes Content-Aggregation- und News-Sammel- und News-Verteil-System bzgl. für die Organisation relevanter Themen an die relevanten Personen. Content Aggregation für interne Wikis/Wissens-Datenbanken/relationale Datenbank-Aufbereitung/eBooks wie auch für Content Marketing (Kunden-Pull; z.B. mit OpenAI GPT-2, Huggingface Transformers).
NLP-basierte Content-Generierung zu den eigenen Beratungsthemen mit GPT-2, GPT-3 und den Huggingface-Tools, d.h. um gefunden zu werden (SEA/SEO) und Anfragen zu erhalten.
Semantische automatisierte E-Mail-Versendung und E-Mail-Beantwortung mit MS Outlook, MS

Exchange und sendmail/postfix/mbox/GNOME Evolution unter Unix sowie vielen NLP-Tools wie Google BERT, Huggingface-Tools, etc.
Insgesamt automatisiertes auf die jeweils relevanten Features und Vorteile abgestimmtes Inbound- und Outbound-Marketing mit NLP/KI sowie Telefonats- und Vertriebs-Gesprächs-Vorbereitung.
Automatisierte Marktanalyse für den Maschinenbau (Lieferketten schließen, neue Lieferanten finden, neue Kunden finden).

UIPath, Progress OpenEdge, Automation Anywhere, blueprism, Jacada, Kofax, Pega, JavaScript, TypeScript, Selenium, KNIME, Docker, Kubernetes, Docker Swarm (managers and workers), MTR, Go/Golang, Java, Scala, Kotlin, Python, Airflow, Kubeflow, TileDB, Couch DB, Uber AresDB, M3DB, YugabyteDB, Apache Hudi, Hazelcast, Soundcloud Roshi, MS Azure Cosmos DB, Redis Enterprise, Apache Atlas (GDPR, Psydonymization), CeleryExecutor, RADOS + Ceph, TensorFlow-Stack mit Keras, AutoKeras oder PyTorch + Auto-PyTorch + AddOns, Uber Horovod, MLFlow, Kedro, Apache Spark Stack mit Spark Streaming, Spark SQL, MLlib, GraphX, Alluxio, TransmogrifAI, Siddhi CEP, Esper + Norikra CEP, TensorFlowOnSpark, PySpark mit Optimus, Apache Flink, Jupyter, Zeppelin, PyTorch, MXNet, Chainer, Keras, Horovod, XGBoost, CatBoost, RabbitMQ, ONNX, Hydrosphere Serving (model management), Zephyr (Continuous Testing Agility), Red Hat OpenShift, Elastic/ElasticSearch, MS Azure Hybrid Cloud, Kafka, Kafka-REST Proxy, KafkaCat, Confluent, Chukwa, Ansible, OpenTSDB, Apache Ignite DB mit TensorFlow/ML-Integration, MLFlow, Kedro, CollectD, Python 3.x, Flask (Python Microframework: REST, UI), Coconut Functional Programming für Python, Robot Framework (Python acceptance test-driven development (ATDD)), CNTLM, Red Hat Identity Manager / FreeIPA, keycloak, Samba, Nginx, Grafana, Jenkins, Nagios, Databricks (Spark, Kafka, Connectors to R, TensorFlow, etc.), OpenCV, Snowflake, RTLinux, RHEL, Ubuntu, AWS, SageMaker, Glue, Scrum + Design Thinking + Value Stream Mapping.

Protokolle: AES, RSA, SHA, Kerberos, SSL/TLS, Diffie-Hellman
DBs: HBase + Phoenix, Hive, PostgreSQL, Druid, Aerospike, Hive, Lucene/Solr/Elasticsearch, SploutSQL
NLP-Stack mit Google BERT/Sling, ALBERT, spaCy, GPT-2, Stanford CoreNLP, AllenNLP, OpenEphyra, DELPH-IN PET Parser, Enju, Grammix, OpenAI GPT-2, Huggingface Transformers.
Bayes- bzw. Stochastik-Libraries / Probabilistic programming (PP) / Programmable Inference: Stan (mc-stan.org), PyMC3/PyMC4, Soss.jl, Julia + MIT Gen, Pyro, Edward on TensorFlow, Microsoft Infer.Net
Probabilistic Logic Networks (PLNs, Pyro-Programmiersprache), Differentiable Programming, Cloned Hidden Markov Models (CHMM).

2/2020 - 3/2020

IT-Architekt, Technologie-Scout und Strategie-Berater, Agile Coach (Freiberuflich)

1. Agile Coaching: SAFe + Design Thinking, Verbesserung der Produktivität und Zusammenarbeit.
2. Erstellung einer speziellen Gesamt-Architektur über die Komponenten und wesentlichen Prozesse in Autonomous-Vehicles (AV), Edge-Processing-Systemen (zur Weiterleitung & Vorverarbeitung der Daten von den Forschungsfahrzeugen für die Verarbeitung im Data Center oder der Cloud. Diese spezielle Gesamt-Architektur diente zur Diskussion mit externen Partnern / Startups über mögliche Kooperationen / Integrationen und Hilfe bei Governance-/Kontroll-Aufgaben, d.h. legt den Schwerpunkt auf hierfür relevante Komponenten und Aspekte während andere geheime firmen-interne Details nicht gezeigt werden.
3. Architektur eines parallelen Data Ingestion Systems von Forschungsfahrzeugen (für autonomes Fahren) ins Datacenter bzw. die AWS-Cloud mit Renovo: Zero-Trust-Architektur mit IAM/IdM und SSO über OAuth und OpenID Connect; Kubernetes & Docker-basiertes cloud-neutrales Hosting.
4. Kubernetes-Docker-basiertes Kosten-Minimierungs-Konzept über AWS-Spot-Instanzen und GPU-Sharing mit Knative, Knative real-time Eventing und kfserving InferenceServices; Start des EKS-Clusters über AWS Lambda. Optionen mit AWS SageMaker für AI-/ML-Tasks sowie AWS Outpost eruiert.
5. Erstellung eines Data-Lake- und Data Vault-Konzepts mit Apache Hudi, Delta Lake, Netflix Iceberg, Parquet/ORC/Avro/Protobuf/ROSbag/ADTF Formate, Ceph + BeeGFS + Lustre Filesystem, MinIO, SkyHook (Query Pushdown), PostgreSQL, InfluxDB, TimeScaleDB, Cassandra, Aerospike. Betrachtete Aspekte: Behandlung von heißen/warmen/kalten Daten, Behandlung von Zeitreihendaten (Zahlen/Zeichenketten), Behandlung von Sensordaten (Blob), Ereignis-/Beschriftungsdaten, die mit Kerndaten verknüpft werden können, Möglichkeit, Metadaten mit Kerndaten zu verknüpfen, welche verschiedenen Arten der Verarbeitung darauf angewandt/durchgeführt werden können, verfügbare/unterstützte Datenformate. Conflict-free Replicated Datatypes (CRDT), Operational Transformation (OT).
6. Analyse der Anforderungen / Algorithmen aus den Methoden des maschinellen Lernens für das autonome Fahren: Sensor-Fusion: Clustering, Segmentierung, Mustererkennung; Straße: Ego-Motion, Bildverarbeitung, Mustererkennung; Lokalisierung: Gleichzeitige Lokalisierung und

Kartierung; Verständnis der (Verkehrs-)Situation: Erkennung und Klassifikation; Planung der Fahraktionen: Bewegungsplanung und –steuerung; Kontrollstrategie: Verstärkungslernen, überwachtes Lernen; Fahrer-Modell: Bildverarbeitung, Mustererkennung.
7. DevOps/AIOps/MLOps/DataOps Konzepte erweitert bzgl. Workflows, Kubernetes/K8s, IAM/IdM
8. AV Data Function- und Data Life Cycle Anforderungsanalyse und Konzept gestartet: Analyse der Datacenter-Use-Cases nach benötigten Daten, Datenlokalität, Nutzung für DAG-Processing mit Apache Spark, der Verarbeitungs- und –Veredelungs-Schritte und wann sie weniger oder irrelevant werden, was wie in der EU pseudonymisiert/anonymisiert werden muss mit Apache Atlas (EU-GDPR, Datenschutz-Grundverordnung DSGVO).
9. Data- und Supplier-Governance und Orchestration/Choreographie-Konzept gestartet: Sicherheits-Kapselungen (Zero-Trust K8s-Container-Anbindung), Kombination der Stärken verschiedener Algorithmen nach Qualitäts- und Safety-Anforderungen, darauf basierend Voting-Architektur für Teilsysteme, Vorschläge zur Kombination verschiedener Startup-Ideen zwecks Synergie-Maximierung.
10. Daten-Suchkonzept mit dem Elastic Stack (Elasticsearch, Logstash, Kibana, Beats) incl. IAM und Data Governance, Zero Trust, Sichtbarkeiten nach Nutzern.
11. Code-Quality Vorgaben erstellt, Toolsets eingeführt und Verbesserungskonzept und Konzeption einer automatisierten CI/CD-Buildkette.

Bibliotheken / Tools
Docker, Argo, ArgoCD, Docker Swarm (managers and workers), MTR, Kubernetes, Scala, Python, Airflow, Kubeflow, TileDB, Couch DB, Uber AresDB, M3DB, YugabyteDB, Apache Hudi, Hazelcast, Soundcloud Roshi, MS Azure Cosmos DB, Redis Enterprise, Apache Atlas (GDPR, Psydonymization), CeleryExecutor, RADOS + Ceph, TensorFlow-Stack mit Keras, AutoKeras oder PyTorch + Auto-PyTorch + AddOns, Uber Horovod, MLFlow, Kedro, Apache Spark Stack mit Spark Streaming, Spark SQL, MLlib, GraphX, Alluxio, TransmogrifAI, Siddhi CEP, Esper + Norikra CEP, TensorFlowOnSpark, PySpark mit Optimus, Apache Flink, Jupyter, Zeppelin, PyTorch, MXNet, Chainer, Keras, Horovod, XGBoost, CatBoost, RabbitMQ, ONNX, Hydrosphere Serving (model management), Zephyr (Continuous Testing Agility), Red Hat OpenShift, Elastic/ElasticSearch, MS Azure Hybrid Cloud, Kafka, Kafka-REST Proxy, KafkaCat, Confluent, Chukwa, Ansible, OpenTSDB, Apache Ignite DB mit TensorFlow/ML-Integration, MLFlow, Kedro, CollectD, Python 3.x, Flask (Python Microframework: REST, UI), Coconut Functional Programming für Python, Robot Framework (Python acceptance test-driven development (ATDD)), CNTLM, Red Hat Identity Manager / FreeIPA, keycloak, Samba, Nginx, Grafana, Jenkins, Nagios, Databricks (Spark, Kafka, Connectors to R, TensorFlow, etc.), OpenCV, Snowflake, RTLinux, RHEL, Ubuntu, Kali Linux, AWS, SageMaker, Glue, Scrum + Design Thinking + SAFe.
Memory-Centric/IMDG (In-Memory Data Grid): Apache Pulsar (schnellere Alternative zu Kafka), memcached, Ignite, GridGain, Alluxio, Redis, Hazelcast, Ehcache, Red Hat JBoss Data Grid, Pivotal GemFire, ActiveMQ, RabbitMQ mit AMQP, MQTT.

Protokolle: AES, RSA, SHA, Kerberos, SSL/TLS, Diffie-Hellman

DBs: HBase + Phoenix, Hive, PostgreSQL, Druid, Aerospike, Hive, Lucene/Solr/Elasticsearch, SploutSQL
NLP-Stack mit Google BERT/Sling, ALBERT, spaCy, GPT-2, Stanford CoreNLP, AllenNLP, OpenEphyra, DELPH-IN PET Parser, Enju, Grammix
Bayes- bzw. Stochastik-Libraries / Probabilistic programming (PP) / Programmable Inference: Stan (mc-stan.org), PyMC3/PyMC4, Soss.jl, Julia + MIT Gen, Pyro, Edward on TensorFlow, Microsoft Infer.Net
Probabilistic Logic Networks (PLNs, Pyro-Programmiersprache), Differentiable Programming, Cloned Hidden Markov Models (CHMM).

4/2019 - 2/2020

Product Owner, fachlicher Projektleiter, agiler Coach (Freiberuflich)

1. Komplettes Aufsetzen des Projekts incl. aller Projektplanung, Personalplanung, Kostenplanung, Erstellung von Projektplänen, Prüfen der CVs, Bewerbergespräche und –Auswahl. Vorstellung des Projekts beim DB-Top-Management und kontinuierlichesReportung.
2. Agile Coaching: SAFe + Design Thinking, Verbesserung der Produktivität und Zusammenarbeit.
3. Requirements Engineering, Use Case 2.0 Engineering der SIEM/SOC Features allgemein und im Bahnkontext. Analyse der Kosten-/Nutzen-Aspekte der Use Cases und deren Abhängigkeiten als Input für agiles Kunden-Wert-basiertes Produktmanagement/Product Owner Tätigkeiten.
4. Recherche, Test und Analyse der führenden Open Source SIEM/SOC Systeme: Apache Metron / HCP (Hortonworks Cybersecurity Platform), Apache Spot, dataShark, Alienvault OSSIM, Graylog, SIEMonster, Hunting ELK (HELK), Wazuh, MozDef, OSSEC, Prelude OSS, Snort, QuadrantSec Sagan, Suricata, OpenStack Vitrage.
5. Splunk: Installation, Konfiguration, Analyse und Anbindung an Input-Quellen, Erstellung von

Splunk-Analyse- und Visualisierungs-Use Cases mit SPL (Search Processing Language).
6. Erstellung einer SOC-Gesamtarchitektur mit Umfängen für Minimal-, Basic-, Advanced- und Premium-Konfiguration mit bis zu 100 Komponenten. Auf dieser Basis Analyse und Präsentation der Chancen/Kosten/Risiken zur Erfüllung von Requirements und Use Cases gegenüber Management und Engineering-Gruppen.
7. Zukunftsvision der SOC-Architektur erstellt auf Basis von Apache Metron + Kafka + Spark + Elastic/ELK Stack (ElasticSearch, LogStash, Kibana) und Konzeption ihrer Komponentenarchitektur - möglichst mit Open-Source-Tools, um Kosten zu sparen. Dazu viele konkrete Vorschläge zur Verbesserung des SOCs (Security Operations Center), Erstellen einer neuen SOC-Architektur mit KI-Elementen: Big Data/Data Science Ansatz zur Angriffs-/Malware-/APT-Erkennung mit Machine Learning und Fokus auf False-Positives-Reduzierung. Visualisierungskonzept zu Angriffs-Verdachtsfällen mit den jeweiligen Security-Kontexten per Design Thinking.
8. Open Source SOC PoC (Proof of Concept): Sammeln der Anforderungen/Use Cases, Erstellung der Architektur basierend auf 3 Säulen: Log-Verarbeitung mit Solr/Elastic, Open Source SOC Elementen (RegEx, Match Expressions mit Spark, Kafka, Solr etc.) sowie einer KI-Säule bestehend aus Data Science und Regel-basierter KI mit Spark sowie Deep Learning mit TensorFlow und PyTorch.
9. Erstellung und Abstimmung des Open Source SOC PoC Projektplans und der Architektur mit dem Top-Management der Bahn (CISO, Technik-Vorstands-Bereich), Erstellung von 7 Job-Profilen und Staffing/Job-Interviews auf dieser Basis.
10. Beschaffung von Deep Learning GPU PC- und Server-Hardware sowie Cloud-Zugängen.
11. Konzeption der Einführung von Docker/Kubernetes für TensorFlow- und PyTorch-Machine-Learning: Vergleich mit der Alternative containerd mit GRPC, Docker Registries mit YAML für Kubernetes, Flannel (layer 3 network config). Kubernetes Tools: kubelet (primary node agent), kube-proxy, Container Runtime, (High Availability) HA endpoints, kubernetes-ha, Kube-apiserver, kubeadm, cluster autoscaler, scheduler, Helm (Kubernetes Package Manager, Microservices), Tiller (Helm server part), Ingress (load balancing, SSL termination, virtual hosting), kube-keepalived-vip (Kubernetes Virtual IP addresses using keepalived), Kubespray (Deploy a Production Ready Kubernetes Cluster). Analyse von Kubernetes & Airflow Failure Stories auf Risiken und Ableitung von Best Practices/Empfehlungen.
12. Auf maximale Performance und Durchsatz optimierte Apache Spark basierende Scheduling-Konzepte mit Alluxio-Caching, Data-Locality-Optimierung und Minimierung datenintensiver Operationen: Custom Spark Scheduler/Spark Task/DAG/SubDAG Combiner für Dynamic Workflows (In-Memory-Optimierungen), Deep Learning Pipelines, Horovod, TensorFlowOnSpark, TensorBoards, TensorFrames, Data Lineage Optimierungen.
13. Erstellung eines umfassenden Testmanagementkonzeptes zur Verbesserung der Stabilität von entwickeltem Code mit den Schwerpunkten Datenaufnahme, KI, DevOps, CI/CD-Pipeline (Continuous Integration/Deployment mit Jenkins und Sonar(Qube)), Metadaten und IT-Sicherheit zur Kanalisierung und Verbesserung von Code durch Developer-Test-, Integrationstest-, Pre-Prod- zu Prod-Umgebungen).
14. Analyse von möglichen Deep Learning Nachfolgetechnologien wie Hierarchical Temporal Memory (HTM), Graph/Memory/Transformer ConvNets (Convolutional Networks) incl. deren frei verfügbaren Implementierungen sowie PLNs (Probabilistic Logic Network): [Naive] Bayesian Belief Networks (BNNs), Markov Logic Networks (MLNs), Conditional Random Fields (CRFs), Direct Graphical Models (DGMs), Statistical Relational Learning (SRL), Stochastic And-Or Grammars (AOGs/SAOGs), Probabilistic Relational Models (PRMs), Markov Logic Networks (MLNs), Relational Dependency Networks (RDNs), Bayesian Logic Programs (BLPs), Probabilistic Graphical Models (PGMs), Markov Random Fields (MRFs), Contextual Graph Markov Models (CGMMs), Hidden Markov Models (HMMs), Human brain neurons (HBNs).
15. Entwicklung eines neuen Explainable AI (XAI) Verfahrens, das Deep Learning ablösen kann durch Verbindung und Weiterentwicklung mehrerer anderer Modelle und Techniken.
16. Förderantrag ausgearbeitet zur Beantragung des Förderprogramms KI-für IT-Sicherheit der Bundesregierung: Innovative Ideen entwickelt, neueste KI-, Data Science und Big Data Verfahren und Weiterentwicklungen vorgeschlagen zur Erkennung von ungewöhnlichem Verhalten/Angriffen/Malware sowie neueste NLP-Verfahren zur automatisierten Analyse von textuellen Angriffs- und Malware-Beschreibungen im Internet oder in E-Mails/Wikis sowie der Umsetzung der Cyber Grand Challenge Elemente über Deep Learning, RNNs, CNNs. Hierzu Entwicklung der Geschäftsstrategie und des Geschäftsplans zur separaten Vermarktung der damit geplanten Innovationen.
17. Erstellen von Sicherheitskonzepten für Windows- und Linux PCs und Sever u.A. durch zahlreiche Sicherheitseinstellungen, mehr Logging, etc. sowie durch Installation von bis zu 50 Analyse- und Überwachungs-Tools wie Sigar, Config. Discovery, File Integrity Checker (Afick), CGC Tools: BinaryAnalysisPlatform bap, angr, s2e, KLEE, Strace, ZZUF, BitBlaze.
18. Konzeption von klassischen Data Science Analysen bzgl. verdächtiger Aktivitäten mit GBM(Gradient Boosting Machine), XGBoost, CatBoost, LightGBM, stacked ensembles, blending, MART (Multiple Additive Regression Trees), Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC.
19. Analyse der besten Deep Learning Netzwerk-Architekturen in den jeweiligen Teilfeldern:

ResNet, ResNext, DenseNet, MSDNet (Multi-Scale DenseNet), RepMet, EfficientNet sowie der folgenden NLP-Implementierungen (z.B. zur Extraktion strukturierter Beschreibungen aus textuellen IoC – Indicators of Compromise): BERT, FastBert, SenseBERT, RoBERTa, GPT, GPT-2.
20. Konzeption/Entwicklung von neuronalen Deep Learning Netzwerk-Architekturen für TensorFlow, Keras, PyTorch mit diesen Elementen: (De-)Convolution, [Dimensional][Min/Max/Average] (Un-)Pooling, Activation Functions, ReLUs (Rectified Linear Units), ELU (Exponential Linear Unit), SELU (Scaled Exponential Linear Unit), GELU (Gaussian Error Linear Unit), SNN (Self Normalizing Network), LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units), Differentiable Associative Memory (Soft RAM/Hash Table), Episodic Memory, Memory Networks, Self-Attention, Multi-Head-Attention, (Masked Multi) Self Attention, NAC (Neural Accumulator), NALU (Neural Arithmetic Logic Unit), Squeeze-and-Excitation (SE) / SENet, SPN (Sum-Product Network), VAE (Variational Auto-Encoders), FCLs (Fully Connected Layers), PLNs (Probabilistic Logic Networks), GANs (Generative Adversarial Networks), Capsule Networks, gcForest, Differentiable Programming, Neural Architecture Search (NAS), Differentiable Neural Networks, [Transposed](De-)Convolutions, ETL (Extract, Transform, Load) with Input/Output Embedding, (Layer) Normalizing, Softmax, Automatic Machine Learning, Episodic Memory, Differentiable Associative Memory, Large Memory Layers with Product Keys, Deep (Double) Q-Learning, (SSL) Semi-/Self-Supervised Learning, Msc (Adding, Concatenation, Segmentation, Linearization, (Convol.) Filters), Reinforcement Learning, Q-learning, Convolutional Models/Learning, Google Dopamine.
21. Konzeption der Deep Learning Architekturen für folgende Use Cases / Use Case Slices: Ausbreitung von Malware durch Security-Zonen, Erkennung des (Check-, Verbreitungs-, Ausleitungs-)Verhaltens von Malware, häufiger Angriffe, insbesondere OS-API-Angriffe, Code Injection, etc., von gestohlenen CPU-Zyklen durch Malware, ggf. durch Hooks in Event-Queues zur Erkennung von deren Abarbeitung, von ROP (Return Oriented Programming) mit ROPNN-Variante auf Standard-Libraries durch Vergleich der üblichen mit den zu beurteilenden Einsprungpunkten; Modelle erstellt für Meta-Level: Netzwerk-Metadaten-Analyse, Detail-Level: Nutzdaten-Analyse auf Exploit-Code/-Daten etc., aktuelle Bedrohungen, bekannt gewordene IoCs, Afick-/tripwire-Daten neuronal analysieren, Erkennung von Verschlüsselung und von Schlüssel-Austauschen.
22. Detail-Vergleich von Elastic mit Solr, der führenden JavaScript-Frameworks: React, Angular und Vue.js, die jeweiligen Native-Frameworks (Ionic etc.) sowie Electron Platform sowie der führenden Clouds: Amazon AWS, Google GCP und Microsoft Azure sowie Docker/Kubernetes, Websockets vs REST, GraphQL vs Odata vs ORDS, Vergleich geeigneter DBs, z.B. für Range-Scans, AWS RedShift vs Athena.
23. Detail-Konzeption der folgenden Solr-Aspekte: SolrCloud/HDP Search, Integration mit Apache Ranger + Sentry + Atlas, Performance-optimierter SolrJ Client mit parallelen Queries, Distributed Indexing, Index Sharding, Shard Splitting und Rebalancing (auch zur Laufzeit), Cross Data Center Replication (CDCR), Solr Security (Kerberos, AD-Anbindung, SASL, SSL), Versionierung mit Avro & LDP (Linked Data Platform) & Apache Marmotta/RFC 7089, Stretched Cluster vs synched Multi-Cluster, Sizing, Definition der Solr Index Identifier (UID), High Availability (HA) und Disaster Recovery (DR) Mechanismen, Solr HA, Load-Balancing-Konzept (HW-basiert über F5, Ping gegen SolrCloud Node, solr healthcheck, Zookeeper, Content-Query gegen Test-Collection, SolrJ Client), Q Replikation, Konzeption von Overlay-Netzen (SDN, Software-Defined Networking).
24. Konzeption der Amazon AWS Cloud-Architektur mit Migrationskonzept in die Cloud und vom monolithischen Ansatz hin zu Microservices, Risiko-Vermeidungsstrategie, Virtualisierung, effizientem JavaScript-UI mit React, Cloud-Sicherheitskonzept, Microservice-Architektur, Microservice-Versionierungsstrategien, optimiertem Datenaustausch, Nutzung des AWS Storage Gateways, AWS Redshift, DDD (Domain-Driven Design) and Bounded Contexts, Product Line Architecture, Single-Sign-On-Konzept (SSO), etc.
25. Recherche und Analyse verfügbarer Sicherheits-Incident- und Hacking-Daten als Input für klassisches Machine Learning (Spark MLlib etc.) sowie für Deep Learning (TensorFlow, PyTorch). Es gibt ca. 100 verschiedene Quellen, aber mit Labeling in unterschiedlicher Qualität, unterschiedlichem Konvertier- und Anpassungsaufwand, etc.
26. Generierung eigener IT-Sicherheits-Trainingsdaten für Machine Learning (ML) über voll-instrumentierte Linux- und Windows-basierte Umgebungen (PC, vmWare), in denen dann ca. 50 PenTesting Tools wie MetaSploit, AutoSploit etc. ausgeführt wurden. Anleitung zur Normalisierung und zum Labeling der so erstellten sowie der externen Daten. Erstellung/Extraktion von regulären Ausdrücken sowie Generierung von ähnlichen Angriffen/Payloads auf dieser Basis.
27. Konzeption+Entwicklung einer Kontroll- und Steuerungs-Library in Scala für Erkennung und KI, die alle Kernelemente des SOCs monitored und steuert.
28. Konzeption+Entwicklung einer UI- und Query-Library in Scala, die intelligente Analysen im Kibana-Dashboard mit React visualisiert sowie nach unten über Apache Drill mit Drillbits Query-Mapping in SQL, HQL, Solr und ähnliche Dialekte durchführt. Hierbei haben wir weitgehend Splunk's SPL (Search Processing Language) als unsere OPL (Open Processing Lanaguage) nachgebildet. Dabei handelt es sich im Wesentlichen um SQL erweitert um Infos zur Darstellung im UI.

29. Recherche/Analyse/Erweiterung aktueller Ideen/Tools zu technischen Knackpunkten in den (Teil-)Projekten oder direkter Vorschlag der Lösungen:
a. Analyse von Semantik-Tools, Symbolic AI und Explainable AI für das KI-Security-Förderprogramm sowie für neue Arbeitspakete: KL-ONE: Protégé, LOOM, Knowledge Engineering Environment (KEE), Pellet, RacerPro, FaCT++ & HermiT, Non-Linear Planner, CBR (Case-Based Reasoning), RDF (Resource Description Framework)/ SPARQL (SPARQL Protocol and RDF Query Language), OpenCog (AtomSpace, Atomese, MOSES/MetaCog, Link-Grammar), Induktions-/Deduktions-Technologie wie OWL/OWL-DL (Ontology Web Language Description Logics), führende Implementierung: Apache Jena OWL, HPSG (Head-driven Phrase Structure Grammar) Parsing: DELPH-IN PET Parser, Enju, Grammix, Stanford CoreNLP, OpenEphyra, Frame-Logik, Explainable AI mit LOCO (Leave-One-Covariate-Out).
b. NLP (Natural Language Processing) / Computerlinguistik Forschung & Auswertung: Analysieren/Parsen natürlicher Szenenbilder zusammen mit dem textuellen Parsen von Bildunterschriften/Beschreibungen aus dem Internet zum Trainieren von Bildverarbeitungsmodellen (Stanford CoreNLP-Ansatz); Klassifizieren von Trouble Tickets / Texten in Kategorien/Aktualitäten; Wartung / Gelernte Lektionen: Analyse textueller Berichte von Technikern über IT-/Fahrprobleme und autonome Fahrtenschwierigkeiten (falsche Klassifizierungen/Reaktionen) für Erkenntnisse/Feedbacks auf NLP-Ebene.
c. Für NLP Generation: OpenAI GPT/GPT-2 (Generative Pre-trained Transformer), Facebook XLM (Cross-lingual Language Model Pretraining), Google BERT (Bidirectional Encoder Representations from Transformers)).
d. KI/AI/Data Science/Big Data: Algorithmen und Tools: LSTM vs. GRU, Feast AI Feature Store, K8s Sidecar Injector, TensorFlow 2.0 (Vorteile von Update/Migration), Tensor Comprehensions, Neural Ordinary Differential Equations, Visual Common Sense Reasoning, Deep Learning, RNNs, CNNs
e. Vorschläge zur Deep-Learning-Beschleunigung u.A. mit aktuellen Publikationen (z.B. Modell-Kompression, Nutzung von HW-Eigenschaften) sowie der Integration von Domänen-Wissen/Semantik/Regeln/Entscheidungstabellen/Ontologien/Erklärbare-KI-Ergebnissen in Deep Learning; Entwicklung von optimierten Hybrid-Learning-Modellen (Deep [Reinforcement] Learning mit klassischen Lernverfahren kombiniert).
f. Konzept für AIops (Artificial Intelligence Operations) / KI-basierte Betriebs-Optimierung im Kontext Metadatamanagement und Ingest:
i. Konzept für die Einführung eines CMS (Config Management System)/ISMS zur Minimierung menschlicher Fehler bei der Programmierung / Ausführung der Skripte: Alle relevanten fest programmierten Parameter wurden in eine separate CMS-Datenbank oder minimal in umgebungsspezifische Konfigurations-/Property-Dateien extrahiert. D.h. ein Parametersatz für die Entwicklungsumgebung, einer für die Testumgebung,.... bis zur Produktionsumgebung (Python NetworkX, Snowflake, …).
ii. Konzept zur Skalierung und Beschleunigung von KI-Workloads, Verwaltung komplexer Workloads, Beschleunigung der Entwicklung und Bereitstellung statistischer Modelle, Voroptimierung in Plattformen für KI-Workloads.
30. NLP-Analyse (Natural Language Processing) von Log- und Web-Inhalten:
a. Extraktion von Fließtext-IoC-Inhalten (Indicator of Compromise) ins STIX-Format zur teilautomatischen Weiterverarbeitung, etwa automatisierte Suche nach Dateihashes, Analyse & Sperren von offenen Ports und ein-/ausgehenden Verbindungen.
b. Semantische Kategorisierung (Problem-Kategorie, Schwere des Fehlers und möglicher Auswirkungen/Risiken, Dringlichkeit) und textuelle NLP-Analyse von Log-Inhalten mit genSim, spaCy und in Teilen auch mit Google BERT, GPT, Graph-ConvNets mit Octavian, Google Sling, TensorFlow graph_nets & gcn (Graph Convolutional Networks), PyTorch Geometric.
c. Data Science-Beratung sowie Management-und Konvertierungskonzepte für Machine-Learning-Modelle mit ONNX (Open Neural Network Exchange?: High-performance optimizer and inference engine for machine learning models and converter between TensorFlow, CNTK, Caffe2, Theano, PyTorch, Chainer formats).

Docker, Argo, ArgoCD, Docker Swarm (managers and workers), MTR, Kubernetes, Scala, Python, Airflow, Kubeflow, CeleryExecutor, RADOS + Ceph, TensorFlow-Stack mit Keras, AutoKeras oder PyTorch + Auto-PyTorch + AddOns, Uber Horovod, Apache Spark Stack mit Spark Streaming, Spark SQL, MLlib, GraphX, Alluxio, TransmogrifAI, Alluxio, TensorFlowOnSpark, PySpark mit Optimus, Jupyter, Zeppelin, PyTorch, MXNet, Chainer, Keras, Horovod, XGBoost, CatBoost, RabbitMQ, ONNX, Hydrosphere Serving (model management), Zephyr (Continuous Testing Agility), Red Hat OpenShift, Elastic/ElasticSearch, MS Azure Hybrid Cloud, Kafka, Kafka-REST Proxy, KafkaCat, Confluent, Ansible, migriert nach SaltStack, OpenTSDB, Apache Ignite DB mit TensorFlow/ML-Integration, CollectD, Python 3.x, Flask (Python Microframework: REST, UI), Coconut Functional Programming für Python, Robot Framework (Python acceptance test-driven development (ATDD)), CNTLM, Samba, Nginx, Grafana, Jenkins, Nagios, Databricks (Spark, Kafka, Connectors to R, TensorFlow, etc.), Snowflake, Data Vault 2.0, FreeRTOS, Zephyr OS, RTLinux, RHEL, Ubuntu, Scrum + Design Thinking + SAFe.
PenTesting-Tools: AutoSploit, Metasploit, Burp Suite, NeXpose, Nessus, Tripwire, CORE Impact, Kali Linux, Snort, Bro, Argus, SiLK, tcpdump, WireShark, parosproxy, mitmproxy,

nmap, Security Onion, Bro, Sguil, Squert, CyberChef, NetworkMiner, Silk, Netsniff-NG, Syslog-NG, Stenographer, osquery, GRR Rapid Response, Sysdig Falco, Fail2Ban, ClamAV, Rsyslog, Enterprise Log Search and Archive (ELSA), Nikto, OWASP Zap, Naxsi, modsecurity, SGUIL, Mimikatz, CORE Impact, Kali Linux.
Log-Processing-Toolsets: OpenSCAP, Moloch, ntopng, Wireshark + plugins, Fluentd Message Parser, SQL-basierte Abfragen: SploutSQL, Norikra + Esper (Stream /Event Processing)
Cyber Grand Challenge (CGC) Tools: BinaryAnalysisPlatform bap, angr, s2e, KLEE, AFL (American fuzzy lop), Strace, ZZUF, Sulley, BitBlaze, Shellphish/Mechaphish Tools: how2heap, fuzzer, driller, rex
Protokolle: AES, RSA, SHA, Kerberos, SSL/TLS, Diffie-Hellman
DBs: HBase + Phoenix, Hive, PostgreSQL, Druid, Aerospike, Hive, Lucene/Solr/Elasticsearch, SploutSQL
NLP-Stack mit Google BERT/Sling, spaCy, GPT-2, Stanford CoreNLP, AllenNLP, OpenEphyra, DELPH-IN PET Parser, Enju, Grammix
Logik-/Semantik-Tools: Protégé, LOOM, RDF (Resource Description Framework)/ SPARQL, OpenCog, Apache Jena OWL, Frame-Logik
OCR/ICR Libraries: Tesseract OCR engine, OCRopus, Formcraft, Kofax KTM (Kofax Transformation Modules)
Sonstige Sicherheits-Tools: IDS/IPS-, NetFlow- und Protokollerfassungs- und Analysetools wie z.B. Snort, Suricata, Bro, Argus, SiLK, tcpdump oder WireShark, Cuckoo-basierte Malware Analyse, Disassembler, Prometheus Monitoring, OCS Inventory NG, System Config + Activity Analysis: Sigar, Config. Discovery, File Integrity Checker (Afick), Apache Nifi / Hortonworks DataFlow, Elastic Stack (Beats, Logstash, Elasticsearch, Kibana, React + Kibana, Solr Stack (SolrCloud, SolrJ Client, Banana), Apache Drill Queries, UIs, Entwicklung von Drillbits, DSL (Domain Specific Language), Eclipse Parser, JavaCC, Antlr, Lex, yacc/bison, Flex, JFlex, GLR/LALR/LL Parser, Ansible, Juju, MAAS, Kubernetes/K8s + Docker, ggf. Minikube, Microk8s, Blitz Incident Response, HDFS, Data Lake, Zookeeper, Hive, JDBC, Management Tools (Ambari, Ranger, etc.), Hadoop Secure Mode, SSO (Single Sign-On), Identity & Access Management (IAM/IdM), LDAP, Role Mapping, Kerberos, TLS, OAuth, OpenId Connect.

11/2018 - 3/2019

KI- und IT-Sicherheits-Architekt und Technical Lead Meta Data Management (MDM) & Ingest, Agile Coach (Freiberuflich)

1. Product Owner, Agile Coaching: Scrum + Design Thinking mit Elementen aus dem Flow-Framework (Project to Product) sowie SAFe-Elementen, Verbesserung der Produktivität, Code-Stabilität und Zusammenarbeit.
2. Strategie zur Fokussierung und Optimierung der agilen DevOps-Team-Performance / Minimierung von Risiken: Die skalierbare Integration Dutzender komplexer teils unreifer Open Source Komponenten ist extrem komplex, weil sie oft je mehrere Hundert Konfigurations-Parameter haben (teils in Config-Files, teils über Aufrufe /Glue Code zu Scripten) und das Job- und Cluster-Situations-bezogen. Zusätzlich sind viele Workarounds oder Fallbacks nötig. Python ist die Risiko-behaftetste Sprache (z.B. weil interpretiert, Fehlerursachen manifestieren sich erst spät, kaum brauchbare Code Quality- oder Refactoring-Tools, wenig etablierte Best Practices, Entwickler kopieren Code von Internet-Trivial-Beispielen und versuchen, damit komplexe Systeme aufzubauen, …). Dann gibt es viele weitere Risiken: Mangelnde Dokumentation, zu wenig kooperative Zusammenarbeit, zu langes Warten auf nötige Inputs/Bottlenecks, zu unvollständig eingeführte Konzepte wie SSO (Single Sign-On) + persönliche Verantwortung, Sicherheits-Features, Logging-/Tracing-Features, stark divergierende wenig wartbare Implementierungen, zu spät bemerkte Limitierungen/Bugs der verwendeten Tools, in der Folge häufiges Umschwenken der Tools, etc.
Entwickelte Lösungsstrategien: Config-Management als Exzellenz-Disziplin + Data Governance / Data Catalogue, AIops (AI Operations), Serverless/Microservices (damit intelligentes automatisches Management und Skalierbarkeit), viele stringente und kontrollierte strategische, taktische und operative Vorgaben aufgrund von Grob-Architektur, Vision und klaren Prioritäten, vollständige Dokumentation, enge effiziente Zusammenarbeit, klare Aufgaben-Verteilung und Planung (strategisches Produkt Management / Portfolio-Management / Produktlinien-Architekturen) mit Berücksichtigung von Abhängigkeiten, Erkennung & Beseitigung von Bottlenecks, intelligentes Monitoring, KI-basiertes Testing (Anomalie-Erkennung in Kombination mit Logging/Tracing) mit mehreren Test-Umgebungen + professionalisierte CI/CD-Pipeline, Code Analyse & Refactorings (Gemeinsamkeiten extrahieren, Utility-Libraries, etc.), Einführung von mehr Code Quality Tools (Analyse/Refactoring/Testing/Tracing/Debugging), Standardisierung/Dokumentation eines jeden neuen Mechanismus (welche Implementierungsvarianten/Tools/Libs/APIs, Namespaces, Stati, Warn- und Fehlermeldungen, welche Diagnose- und Fallback-Mechanismen, Scheduling/Workflow mit strategischer Planung aller Ressourcen und Vermeidung von Deadlocks/Race Conditions, IT-Sicherheit), Erfassung und Nutzen aller Abhängigkeiten (zum Betriebssystem, zu sonstigen Tools/Libs), Definition + Implementierung von Workarounds zu Standard-Problemen wie Stale File Handles, Stale Sockets, Vermeidung von Out-of-X-Meldungen und Thrashing, Netzwerk-Problemen, etc.
3. Security-Konzept für Docker/Kubernetes/K8s: kubectl, Docker Authentication on Kubernetes

pods, AuthN/AuthZ Methods wie UMA 2.0 (Federated Authorization for User-Managed Access), OpenID Connect mit keycloak über Translations, Kubernetes RBAC & User Impersonation, Volume Type Whitelisting, SELinux/seccomp/AppArmor, System Call Filter, Kubernetes Helm Sicherheitslimits & Verbesserungen, DEX vs Keycloak, SSSD PAM module (POSIX) für MapR Filesystem/HDFS, MapR Container Location Database (CLDB), etc.
4. Zukunftsvision der SOC-Architektur erstellt auf Basis von Apache Metron + Kafka + Spark + ELK (Elastic, LogStash, Kibana) und Konzeption ihrer Komponentenarchitektur - möglichst mit Open-Source-Tools, um Kosten zu sparen. Dazu viele konkrete Vorschläge zur Verbesserung des SOCs (Security Operations Center), Erstellen einer neuen SOC-Architektur mit KI-Elementen: Big Data/Data Science Ansatz zur Angriffs-/Malware-/APT-Erkennung mit Machine Learning und Fokus auf False-Positives-Reduzierung. Visualisierungskonzept zu Angriffs-Verdachtsfällen mit den jeweiligen Security-Kontexten per Design Thinking.
5. Vorschlag von Architekturen / Verbesserungen: Zero-Downtime-Architekturen, schnelleres Dateneinlesen, Autonomes-Fahren-Analysierer / robotic-drive analyzer (RDA), Messaging/Workflow und Containerisierungsarchitekturen.
6. Konzeption der Microservices/APIs, u.A. für die Metadatenverwaltung, Machine Learning Parameter, ...
7. Optimierung der Real-time Data Ingestion Verfahren für hochauflösende Self-Driving Car Video- und Sensor-Daten (TB-PB Datenmengen) in einen MapR Hadoop Datalake mit MapR-DB und Ceph Storage (Reliable Autonomic Distributed Object Store (RADOS)), etcd (distributed key value store) mit LoadBalancer (LB), Real-Time Monitoring mit Prometheus und Elastic/ELK.
8. Konzeption der Einführung von Docker/Kubernetes für TensorFlow-MachineLearning: Vergleich mit der Alternative containerd mit GRPC, Docker Registries mit YAML für Kubernetes, Flannel (layer 3 network config). Kubernetes Tools: kubelet (primary node agent), kube-proxy, Container Runtime, (High Availability) HA endpoints, kubernetes-ha, Kube-apiserver, kubeadm, cluster autoscaler, scheduler, Helm (Kubernetes Package Manager, Microservices), Tiller (Helm server part), Ingress (load balancing, SSL termination, virtual hosting), kube-keepalived-vip (Kubernetes Virtual IP addresses using keepalived), Kubespray (Deploy a Production Ready Kubernetes Cluster). Analyse von Kubernetes & Airflow Failure Stories auf Risiken und Ableitung von Best Practices/Empfehlungen.
9. Scheduling-Konzepte mit Airflow, LocalExecutor, Celery (Distributed Task Queue), CeleryExecutor, RabbitMQ, Dynamic Workflows mit DAGs/SubDAGs mit PythonOperator/BashOperator, upstream/downstream/X-COM, Backfill, Catchup, Kubeflow, Seldon Core.
10. Parallelisierung/Optimierung/Skalieren/Wiederaufsetzen/Fortführen von Deep Learning und speziell TensorFlow-Pipelines und supervised Optimierungszyklen, u.A. mit Spark: Horovod (Training + HorovodEstimator für TensorFlow, Keras, and PyTorch), TensorFlowOnSpark, TensorBoards, TensorFrames.
11. Auf maximale Performance und Durchsatz optimierte Apache Spark basierende Scheduling-Konzepte mit Alluxio-Caching, Data-Locality-Optimierung und Minimierung datenintensiver Operationen: Custom Spark Scheduler/Spark Task/DAG/SubDAG Combiner für Dynamic Workflows (In-Memory-Optimierungen), Deep Learning Pipelines, Horovod, TensorFlowOnSpark, TensorBoards, TensorFrames, Data Lineage Optimierungen.
12. Review aller Security-Aspekte: Airflow, Kubernetes, Docker, Zeppelin, Spark, Java-Sicherheit mit Apache Shiro/Spring Security, sichere Speicherung von Anmeldeinformationen im Unix-Dateisystem, Github, Soft/Hard PSE (Personal Security Environment) mit z.B. SSO (Single Sign On with CA SiteMinder, PAI, OpenId Connect), CyberArk PW Vault API, SSO oder GPG + Ansible Vault, etc.
13. Hilfe/Review bei Angular-basierten Visualisierungen, insbesondere für Grafana (zunächst in Angular, dann in React weil Grafana von Angular auf React migriert wurde).
14. Erstellung eines umfassenden Testmanagementkonzeptes zur Verbesserung der Stabilität von entwickeltem Code mit den Schwerpunkten Datenaufnahme, KI, DevOps, CI/CD-Pipeline (Continuous Integration/Deployment mit Jenkins und Sonar(Qube)), Metadaten und IT-Sicherheit zur Kanalisierung und Verbesserung von Code durch Developer-Test-, Integrationstest-, Pre-Prod- zu Prod-Umgebungen).
15. Förderantrag ausgearbeitet zur Beantragung des Förderprogramm KI-für IT-Sicherheit der Bundesregierung: Innovative Ideen entwickelt, neueste KI-, Data Science und Big Data Verfahren und Weiterentwicklungen vorgeschlagen zur Erkennung von ungewöhnlichem Verhalten/Angriffen/Malware sowie neueste NLP-Verfahren zur automatisierten Analyse von textuellen Angriffs- und Malware-Beschreibungen im Internet oder in E-Mails/Wikis sowie der Umsetzung der Cyber Grand Challenge Elemente über Deep Learning, RNNs, CNNs. Hierzu Entwicklung der Geschäftsstrategie und des Geschäftsplans zur separaten Vermarktung der damit geplanten Innovationen.
16. Recherche/Analyse/Erweiterung aktueller Ideen/Tools zu technischen Knackpunkten in den Projekten für den Lieferanten DXC und Weitergabe an den DXC-Vertrieb zur Akquise neuer Arbeitspakete oder direkter Vorschlag der Lösungen samt passenden Autonomous-Driving-Use-Cases an die relevanten Ansprechpartner in den Teilprojekten:
a. Analyse von Semantik-Tools, Symbolic AI und Explainable AI für das KI-Security-Förderprogramm sowie für neue Arbeitspakete: KL-ONE: Protégé, LOOM, Knowledge Engineering Environment (KEE), Pellet, RacerPro, FaCT++ & HermiT, Non-Linear Planner, CBR

(Case-Based Reasoning), RDF (Resource Description Framework)/ SPARQL (SPARQL Protocol and RDF Query Language), OpenCog (AtomSpace, Atomese, MOSES/MetaCog, Link-Grammar), Induktions-/Deduktions-Technologie wie OWL/OWL-DL (Ontology Web Language Description Logics), führende Implementierung: Apache Jena OWL, HPSG (Head-driven Phrase Structure Grammar) Parsing: DELPH-IN PET Parser, Enju, Grammix, Stanford CoreNLP, OpenEphyra, Frame-Logik.
b. NLP (Natural Language Processing) / Computerlinguistik Forschung & Auswertung: Analysieren/Parsen natürlicher Szenenbilder zusammen mit dem textuellen Parsen von Bildunterschriften/Beschreibungen aus dem Internet zum Trainieren von Bildverarbeitungsmodellen (Stanford CoreNLP-Ansatz); Klassifizieren von Trouble Tickets / Texten in Kategorien/Aktualitäten; Wartung / Gelernte Lektionen: Analyse textueller Berichte von Technikern über IT-/Fahrprobleme und autonome Fahrtenschwierigkeiten (falsche Klassifizierungen/Reaktionen) für Erkenntnisse/Feedbacks auf NLP-Ebene; Generieren von a) Beschreibungen für Fahrer, welche Art von Trainings-Situationen im Straßenverkehr anzustreben sind, b) Um welche Art von Fehlerursachen es sich bei gegebenen Symptomen handeln könnte als Liste oder Text.
Tools/Algorithmen: OpenAI GPT-2 (Generative Pre-trained Transformer), Facebook XLM (Cross-lingual Language Model Pretraining), Facebook PyText (NLP Modeling Framework, auf PyTorch), Google BERT (Bidirectional Encoder Representations from Transformers), Kombinierte Multi-Task-Modell-NLP, Vortraining kompletter (Sprach-/Tiefenlernen) Modelle mit hierarchischen Darstellungen, Aufmerksamkeitsmodelle, DLNLP (Deep Learning NLP: Embed, Encode, Attend, Predict), Hierarchical Multi-Task Learning Model (HMTL), semi-supervised Lernalgorithmen zur Erstellung von Proxy-Labels auf unmarkierten Daten, BiLSTM, SalesForce MetaMind-Ansatz, DeepMind, Deep Transfer Learning for NLP, vortrainierte Sprachmodelle, Worteinbettungen / Worttaschen, Sequenz-zu-Sequenz-Modelle, Gedächtnis-basierte Netzwerke, Gegensätzliches Lernen, Verstärkungslernen, semantische Rollenkennzeichnung, Repräsentationslernen, Textklassifizierung mit TensorFlow Estimatoren, word2vec, Vektor-Raum-Modell/Mapping von Features zu Einbettungen, Skip-Grammen, Seq2seq Encoder-Decoder, ULM-FiT, ELMo, OpenAI Transformer / GPT, Google BERT, BERT, Transfer Learning, OpenAI Transformer, spaCy + Cython zur Beschleunigung, OpenNMT (Neural Machine Translation), AllenNLP (auf PyTorch), OpenNLP, Verstärkungslernen zum Erlernen korrekter Klassifizierungen/Labelzuweisungen/Fragen & Antworten, tief latente Variablenmodelle, Visual Commonsense Season Reasoning, Modell-agnostisches Meta-Learning (MAML), Multi-Hop-Denken, Aufmerksamkeitsmasken für (Self-Attention) GANs (SAGAN), TensorFlow Lingvo (NLP sequence models), OpenEphyra (Teil von IBM Watson).
c. Für NLP Generation: https://blog.openai.com/better-language-models/ (Interesting technologies: OpenAI GPT-2 (Generative Pre-trained Transformer), Facebook XLM (Cross-lingual Language Model Pretraining), Google BERT (Bidirectional Encoder Representations from Transformers)).
d. KI/AI/Data Science/Big Data: Algorithmen und Tools: LSTM vs. GRU, Feast AI Feature Store, K8s Sidecar Injector, TensorFlow 2.0 (Vorteile von Update/Migration), Tensor Comprehensions, Style GANs, Neural Ordinary Differential Equations, Visual Common Sense Reasoning, Deep Learning, RNNs, CNNs for Self-Driving Cars / Logically/temporally consistent virtual 3D city generation, Deep Labelling for Semantic Image Segmentation mit Keras/TensorFlow, Design Patterns for Deep Learning, RNN, CNN Architectures, DeepMind (Kapitan, Scalable Agent, Learning to Learn, TF Reinforcement Learning agents), Uber's QALM (QoS Load Management), Fusion.js (JS framework supporting React, Redux & pre-configured optimized boilerplate, hot module reloading, data-aware server-side rendering, bundle splitting, plugin-architecture, observability, I18n), Horovod (distributed training framework for TensorFlow, Keras, PyTorch), Ludwig (train and test deep learning models without coding), AresDB (Uber's GPU-powered real-time analytics engine), Uber's Sparse Blocks Network (SBNet, TensorFlow algorithm), Google Dopamine reinforcement learning framework based on TensorFlow, Kubernetes Operator für Apache Spark, FastAI Deep Learning, Polygon-RNN++, Flow Framework: Project to Product Agile Process, IntelAI OpenVINO (inference serving component for AI models), IntelAI Nauta (distributed computing environment for running DL model training), TensorFlow Extended (TFX), Salesforce Einstein TransmogrifAI (machine learning automation with AutoML), OpenCV (Open Computer Vision Library), GluonCV, Angel-ML (handling higher dimension ML models), Acumos AI (design, integration and deployment of AI models; AI Model Marketplace), (Paddle EDL: Elastic Deep Learning framework: optimizes deep learning job and waiting time in the cluster: Kubernetes controller & fault-tolerable deep learning framework: PaddlePaddle & TensorFlow), Pyro (Deep Probabilistic Programming Language), Jaeger (OS distributed tracing system, optimized for microservices).
e. Vorschläge zur Deep-Learning-Beschleunigung u.A. mit aktuellen Publikationen (z.B. Modell-Kompression, Nutzung von HW-Eigenschaften) sowie der Integration von Domänen-Wissen/Semantik/Regeln/Entscheidungstabellen/Ontologien/Erklärbare-KI-Ergebnissen in Deep Learning; Entwicklung von optimierten Hybrid-Learning-Modellen (Deep [Reinforcement] Learning mit klassischen Lernverfahren, Regeln, Constaints, Tabellen kombiniert).
f. Machine Learning / Image / Video-Analyse-Tool Recherche und Integrationskonzepte für Sensor Fusion, sonstige Daten-Zusammenführung, Massendatenverarbeitung, UML-Software-Architektur: OpenCL (Computing Language für div. HW Plattformen), OpenCV (Computer

Vision), OpenVX (Vision Cross-Platform), Vulkan, OpenGL (ES), CUDA, nVidia GPU Toolkits wie VulkanRT.
g. Konzept für AIops (Artificial Intelligence Operations) / KI-basierte Betriebs-Optimierung im Kontext Metadatamanagement und Ingest:
i. Konzept für die Einführung eines CMS (Config Management System) zur Minimierung menschlicher Fehler bei der Programmierung / Ausführung der Skripte: Alle relevanten fest programmierten Parameter wurden in eine separate CMS-Datenbank oder minimal in umgebungsspezifische Konfigurations-/Property-Dateien extrahiert. D.h. ein Parametersatz für die Entwicklungsumgebung, einer für die Testumgebung,.... bis zur Produktionsumgebung (Python NetworkX, Snowflake, …).
ii. Konzept zur Skalierung und Beschleunigung von KI-Workloads, Verwaltung komplexer Workloads, Beschleunigung der Entwicklung und Bereitstellung statistischer Modelle, Voroptimierung in Plattformen für KI-Workloads: Datenaufnahme und -aufbereitung, Datenmodellierung und -schulung, Datenbereitstellung und -betrieb, Integration von maschinellem Lernen mit vorgefertigten Blueprints für Chef/Puppet/Ansible/Airflow, automatisierte Speicherkapazitätsbereitstellung, vorausschauende Speicheroptimierung (in hyperkonvergierten Umgebungen), KI, die hyperkonvergierte Hardware zur Anwendungsbeschleunigung konfiguriert, Passwort und "PII-Discovery" (PII = Personally Identifiable Information), wann Lasten mit hohen CPU-/GPU-Anforderungen und -Nutzungsdauern zu starten sind (die z.B. zu Deadlocks/Timing-Problemen oder dazu führen können dass andere Jobs warten müssen), wann Deep Learning/KI-Jobs mit geringerer Priorität zu starten sind und wann Ressourcen auf hochpriore Jobs/Lasten verschoben werden müssen, wann Diagnostik-Sammelprozesse nach Warnungen/Fehlern/Ausfällen gestartet werden, …
h. Vorschlag, Ausarbeitung und Diskussion der geplanten/angebotenen Arbeitspakete zu Techniken, Tools und Innovationen mit Automobilherstellern und anderen Kunden.
i. Data Science-Beratung sowie Management-und Konvertierungskonzepte für Machine-Learning-Modelle mit ONNX (Open Neural Network Exchange?: High-performance optimizer and inference engine for machine learning models and converter between TensorFlow, CNTK, Caffe2, Theano, PyTorch, Chainer formats).
TensorFlow für Bild-/Video-Analyse: Labeling und überwachtes Lernen zur korrekten Klassifizierung, verteiltes Hyper-Parameter-Tuning mit TensorFlow, Keras. ML Debugging/Erklärbare KI im Kontext von LIME, SHAP, partielle Abhängigkeitsdiagramme[Modellleckagen, Entscheidungserklärungen in if-Anweisungen, ...]; Modellspeicherung in PMML mit OpenScoring.io und HBase/MapR-DB + Apache Phoenix. MapR Hadoop, MapR-DB, MapR Control System (MCS) , MapR POSIX Clients, MapR expandaudit, Mesos, Hive, Ceph, RADOS, TensorFlow, Apache Spark, Alluxio, TensorFlowOnSpark, PySpark mit Optimus, Docker, Kubernetes, Airflow, Kubeflow, CeleryExecutor, Jupyter, Zeppelin, PyTorch, MXNet, Chainer, Keras, Horovod, XGBoost, Keras, PyTorch, RabbitMQ, ONNX, Hydrosphere Serving (model management), Zephyr (Continuous Testing Agility), Red Hat OpenShift, Elastic/ElasticSearch, MS Azure Hybrid Cloud, Kafka, Kafka-REST Proxy, Confluent, Ansible, migriert nach SaltStack, OpenTSDB, Apache Ignite DB mit TensorFlow/ML-Integration, CollectD, Python 3.x., Flask (Python Microframework: REST, UI), Coconut Functional Programming für Python, Robot Framework (Python acceptance test-driven development (ATDD)), DaSense 2&3, DaSense GPU Scheduler, CNTLM, Samba, Nginx, Grafana, Jenkins, Nagios, eXtollo Plattform (Azure HDInsight, Azure Keyvault, Azure Databricks (Spark, Kafka, Connectors to R, TensorFlow, etc.)), Snowflake, Data Vault 2.0, FreeRTOS, Zephyr OS, RTLinux, RHEL, Ubuntu, Simulationstechniken wie Hardware-in-the-loop (HIL) and Software-in-the-loop (SIL), Light detection and ranging (LiDAR), Automotive Data and Time-Triggered Framework (ADTF), ROSbag (Robot Operating System – ROS bag), Main distribution frame 4 (MDF4), Scrum + Design Thinking + SAFe + Flow-Framework (Project to Product).

12/2017 - 11/2018

Product Owner, IT Architekt (Freiberuflich)

Tätigkeit: 1. Konzeption der Security-Maßnahmen für das neue SAP Core Banking System als Security Architect.
2. Überprüfung von Use Cases auf Relevanz für DSGVO/Datenschutz und Erstellung entsprechender Bewertungen, Ausfüllen von DSGVO-Formularen.
3. IAM (Identity and Access Management): SAP NetWeaver Identity Management (IdM) eingeführt mit SAML, OAuth, OpenId Connect, Kerberos; Konsolidierung der IAM-/IdM-Funktionalität, die vorher über verschiedene Technologie-Inseln verteilt waren wie LDAP, Active Directory (AD) Federation Services (ADFS), RACF, Oracle Enterprise Directory Server (OEDS), Lotus Notes Domino, etc.
4. Vorschlag von abgeleiteten IT-Security-Architektur- und DSGVO-Maßnahmen auf Basis der vorhandenen Grob-Architektur, Konzept für Privileged Account Management (PAM) und weitergehende Sicherheits-Maßnahmen.
5. Zukunftsvision der SOC-Architektur und Konzeption ihrer Komponentenarchitektur - mit möglichst vielen Open-Source-Tools, um Kosten zu sparen und neuesten KI/AI (Künstliche/Artificial Intelligence) und Machine Learning Frameworks: Spark + MLLib, XGBoost, ….

6. (Weiterer) Aufbau des SOCs (Security Operations Center) als Architekt/PM mit am Ende ca. 60 Security-Tools. Davon wurden ca. 15 Tools neu eingeführt. Deren Einführung sowie die Integration und Automatisierung eines Großteils der Tools habe ich insbesondere konzipiert und in Teilen programmiert: Automatisierte Echtzeit-Datenflüsse und Reduktion von False-Positives.
7. Red-Blue-Team Testing / Penetration Testing / PenTesting und Verteidigung, insbesondere bzgl. der Verwundbarkeit gegenüber aktuellen Exploits und den Indikatoren im SIEM und den Folgen/Risiken für die IT und der Optimierung der möglichst schnellen Erkennung mit wenigen False Positives.
8. Evaluierung der Risk Management Frameworks IRAM2, FAIR, OCTAVE, COSO gegen den MaRisk-Standard von 2017 und BAIT (Bankaufsichtliche Anforderungen an die IT).
9. Erweiterung und Umsetzung von Vulnerability Management, Patch Management und Security-Standards-Compliance sowie Dokumentation dazugehöriger Risiken.
10. Patching-/Risk-Projektmanager Germany bzgl. Meltdown/Spectre (CPU Bugs).
11. Mitarbeit bzgl. IT-Sicherheit an der R3/Corda Blockchain Implementierung der HSBC in Kotlin mit über 100 anderen Banken und Vorbereitung der Herausgabe des Utility Settlement Coins (USC) der Großbanken sowie der Anbindung der Big Data basierenden Bank-eigenen Fraud Detection Lösung, z.B. bzgl. Security-Anbindung per BlueTalon + Ranger.
12. Integration von Security-Systemen per Serverless-Architektur über Google Cloud Functions per REST APIs mit Go: Automatisierte Integration von Configuration Management, Nessus- + Tripwire-Security Scans (Windows/Linux Datenbanken: Verwundbarkeiten und Compliance-Einstellungen) sowie der datenbankbasierten Auswertung der Scans (manuelle Gewichtungen) und Weiterleitung/Eskalation der Ergebnisse.
13. Mitentwicklung von Mobile-App- und Cloud Security Standards, insbesondere für Hybrid Clouds mit dem Google Cloud Stack, z.B. der Software-Defined Perimeter Ansatz.
14. Architektur obiger APIs nach Open Banking Standard mit Mulesoft AnyPoint Platform (API Gateway, App execution, API Repository & Portal, API Designer, Runtime Manager, CloudHub, Private Cloud, AnyPoint Studio).
15. Beratung der Architekten und Entwickler-Teams bzgl. sicherer Konzeption/Entwicklung, sicherer Anbindung von Security Libraries (z.B. Spring Security, SAML, OAuth, LDAP, OpenId Connect), Patchen von Library-Verwundbarkeiten (Vermeiden/Minimieren der Verwendung von anfälligen Versionen: Lösungen und Workarounds) und Security Code-Review mit Tool-Unterstützung (ConQAT + Teamscale von CQSE, Support Query Framework (SQF) und Code Inspector von SAP (ABAP), Micro Focus Fortify, LGTM, Semmle, FindBugs, PMD, SonarQube, Checkstyle, etc.) im Rahmen von TQE (Total Quality Engineering).
16. Beratung bei der Weiterentwicklung der Asset Management und Configuration Management Datenbanken/Systeme um priorisierte Risiko- und Gegenmaßnahmen-Einschätzung in Richtung des statistischen Common Criteria Ansatzes.
17. Internal Reviews/Assessments, Erstellen von Management Self-Identified Issue (MSII) Berichten als Vorbereitung für offizielle Reviews/Assessments.
18. Business Impact Analysis (BIA) und Global Application Security Risk Assessments (GASRA).
19. Business Process Definition / Optimization / Re-Engineering: Network Based Intrusion Prevention (NIPS), Vulnerability Management, Privileged Access Management, Testing & Patching, Anlegen/Anpassen von Beantragungs-/Entziehungs- und Überwachungsprozes-sen mit Neocase Advanced BPM Suite / NEO Process Manager.
20. Security-Architektur für einen Amazon-Cloud- und Serverless-PoC: AWS, Fargate, S3, EC2, VPC (Virtual Private Cloud), IAM, RDS, RedShift, Aurora, DynamoDB (Rel. DBs), Neptune (Graph DB), ElastiCache (In-Mem-DB), Elastic Beanstalk (Orchestration Srv), CloudTrail (Sec. Log), STS (Secure Token Srv), EKS (Elastic Kubernetes Service), EBS (Elastic Block Store), OpsWorks (Config Mgmt), SQS (Simple Queue Srv), CloudWatch (Billing/Metrics), Docker, Kubernetes, Kubeless, Go.
21. Security-Architektur für PoCs mit Blockchain for trade (We.Trade, Voltron, R3/Corda), Biocatch, Microplatforms, Eclipse Microprofile (Hammok, Red Hat Wildfly Swarm, Open Liberty/WebSphere Liberty), JWT, OpenTracing, MicroNaut, ThreatMetrix, UNSilo, Skytree, TidalScale, DataRobot, data iku, Ayasdi (AML), Quantexa, Seldon.io, gVisor.
22. Unterstützung bei der Einführung agiler Prozesse: Design Thinking (Empathie-Maps, Personas, User Profile Canvas, Value Proposition Canvas, Business Model Canvas, Business Ecosystem Canvas, Customer Journeys, HOOK (Trigger, Action, Variable Reward, Investment), SCAMPER (Substitute, Combine, Adjust, Modify, Put to other uses, Eliminate, Rearrange), MVP, MVE (Minimum Viable Ecosystem), Virtuous Loops, Systems Thinking, Business Ecosystem Design, Lean Canvas, NABC (Needs Approach Benefits Competition), SWOT) in Kombination mit DAD (Disciplined Agile Delivery) und SAFe (Scaled Agile Framework) – insbesondere Coaching und Halten von Präsentationen zu den Risiken agiler Verfahren – u.A. durch das Entfallen der Architektur-Phase (siehe meine Social Media Accounts), Mit-Einführen von WorkHacks (= LifeHacks für den Beruf).
23. Konzeption + (Teil-)Implementierung einer automatisierten Microservice/Serverless System-Security- und Vulnerability-Assessment und Reporting-Komponente in Python3 und JavaScript (mit PhantomJS, CasperJS, Bootstrap, a2ps), die automatisiert HTML- und PDF-Reports erzeugte aus Statistical Common Criteria Bewertungsergebnissen, Nessus- + Tripwire-Scan-Ergebnissen, CMDB-Infos (Config Mgmt DB namens ITDoku) etc. mit Integration zu

diversen Systemen (Lotus Notes, CMDB, Excel-Dateien, Oracle-DB, CyberArk Password Vault, Inventory-Systemen zum Check der Kritikalität (BIA/GASRA), Installationsstatus von Security-Tools, etc.) per REST APIs, SysCalls und OAuth.
24. Insgesamt ca. 50 Verbesserungsvorschläge unterbreitet/umgesetzt, vor allem zur Verbesserung des SOCs / der effizienten Erkennung, Priorisierung und Beseitigung von Risiken/Angriffen.
25. Erstellung/Erweiterung/Schärfung von ca. 150 QRadar SIEM Use Cases für zielgerichteteres Security-Monitoring mit weniger False Positives oder weniger manuellem Nachrecherche-Bedarf bei Alerts (Minimierung der manuellen Aufwände).
26. SIEM-Alternativen: Evaluation von
a. ElasticSearch + Norikra Schemaless Stream Processing + Esper CEP (Complex Event Processing) + Apache Nifi + Kafka + Fluentd für SIEM Use Cases/Alerting, Datenextraktion aus Protokollen per WireShark-Plugins (z.B. bzgl. SMBv1 + v2 Exploits [EternalRomance, EternalBlue, EternalChampion, WannaCry]),
b. Apache Metron (ex: Cisco OpenSOC) + Blitz Incident Response + Apache Nifi + Hadoop + Apache Solr/HDP Search + Ranger + Atlas, Technologie-Workshops. Konzeptionen zu:
i. Dokumenten-Id-Vergabe und expliziter Verteilung der Dokumente auf Shards/Replicas und dessen Tracking.
ii. Parallelisiertem SolrJ-Client optimiert auf Antwort-Geschwindigkeit.
iii. Loadbalancer-Switching-Logik.
iv. Schutz gegen bösartige Ambari-Administratoren.
v. Integration der Lösung in das Single Sign On (SSO) Konzept mit Identity & Access Management per LDAP, SASL, explicit TLS.
27. Konzeption/Implementierung eines Apache Spark + MLlib + Kafka basierenden Data Science und Machine Learning Systems zur Erkennung von Incidents/Malware/Netzwerk Anomalien mit H2O.ai.
DS-Ansatz (Data Science) zur Erkennung von Incidents/Malware/Netzwerk-Anomalien Eine Mischung aus Hauptkomponentenanalyse, Nearest Neighbor Methoden, neuronale Netze, Zeitreihenanalyse, Anomalie-Erkennung, Assoziationsanalyse, Maximum-Likelihood-Schätzer, Random Forest, Gradient Boosting (GBM(Gradient Boosting Machine), XGBoost), CatBoost, LightGBM, SHAP (SHapley Additive exPlanations), stacked ensembles, blending, MART (Multiple Additive Regression Trees), AutoML, Auto-Keras, Dopamine, Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC, Cubist (Erweiterung von Quinlan's M5 model tree), C4.5, Assoziationsanalyse, (Nicht)lineare Regression, Multiple Regression, Apriori-Analyse, Überwachte Klassifizierung, Link-Analyse-Netzwerke.
Bibliotheken / Tools SAP Basis, FI/CO, DM, CM, LM, FSCM, FS, FS-BA, SAP NetWeaver Iden-tity Management (IdM), IBM FileNet, SAP Business Objects, Mulesoft AnyPoint Platform (API Gateway, App execution, API Repository & Por-tal, API Designer, Runtime Manager, CloudHub, Private Cloud, AnyPoint Studio), Symantec DCS, Symantec DLP, Symantec PGP Server, Syman-tec SSLVA, TrendMicro Deep Discovery + Antivirus (AV), Cisco Router, ASA, Switches, CheckPoint Firewalls/IDS/IPS, Barracuda WAF, Windows & SAP PKI & IAM, IBM QRadar, IBM Resilient, IBM InfoSphere Guardium (Monitoring: DB, etc.), IBM Vanguard, IBM RACF, IBM EventAction, Nessus Vulnerability-Scanner, ForeScout (vulnerable IoT), Proofpoint (E-Mail Security), CrowdStrike (Endpoint Protection), McAfee (Antivirus + HIPS + Drive Encryption + E-Mail Gateway + ePolicy Orchestrator ePO), Skyhigh (Web Browser isolated in the Cloud, Secure Cloud Services), MenloSecurity (DLP, Absichern von E-Mail- und WebLinks), Cisco Open DNS, BlueCoat Proxy/SSL Decryption/AV, CyberArk Password Vault + Privileged Threat Analytics (PTA), Tufin (Network Security Policy + Firewall Management), Ivanti Application Control (ex: AppSense), En-case Endpoint Security/Forensics, Lumension Endpoint Security, Micro-soft Baseline Security Analyzer (MBSA), RSA enVision, SCCM Windows Compliance, Trustwave DbProtect, DB SAT, Avecto Defendpoint, Centrify DirectAudit, Dark Trace (UEBA: User & Entity Behavior Analytics / NGAV: Next-generation antivirus platforms / DER: Endpoint Detection and Response), DFLabs (SOAR: security orchestration, automation and response), AutoSploit, MetaSploit, Cuckoo Malware Analysis (in virt. Sandbox), MS Visual Studio, Eclipse + Java 1.8, Keycloak, Snort, Python 3.7, p0f, Cluster SSH, Open Workbench, viele Open Source Tools (Fuz-zer, Exploits, Utilities, …), Vizolution, Google Cloud Platform (GCP: Cloud Functions/Datastore/Storage, Cloud Pub/Sub, Endpoints, RSocket, Tools: gVisor (User Space Kernel), Apigee, Cloud Dataflow, BigTable, BigQuery (DWH), BigQuery ML (BQML), Firestore, Firebase, Memo-rystore, Datastore, Cloud Spanner, Cloud Launcher, Cloud SQL, BigCom-pute, Cloud ML Engine, Apache Beam, bdutil, Dataproc (Managed Ha-doop), Stackdriver (Systems Management), AutoML, Google Kubernetes Engine (GKE)), Apache Spark + MLlib + Kafka, H2O.ai, We.Trade, Volt-ron, R3/Corda), Biocatch, Microplatforms, Eclipse Microprofile (Ham-mok, Red Hat Wildfly Swarm, Open Liberty/WebSphere Liberty), JWT, OpenTracing, MicroNaut, ThreatMetrix, UNSilo, Skytree, TidalScale, Da-taRobot, data iku, Ayasdi (AML), Quantexa, Seldon.io, gVisor.

9/2017 - 12/2017 **Coach: Big Data Architektur & Data Science (Freiberuflich)**

1. Zwecks Einarbeitung & Coaching-Grundlage: Erhebung der Ist-Situation bzgl. Tools, Algorithmen und IT-Umgebungen; Mitarbeit bei der Erstellung von Ab Initio Graphen/Lineages als ETL-Pipelines unter Integration von Teradata BTEQs/ActiveBatch/SQL, R, Python, Spark, Hive, SAP, MicroStrategy.
2. Big Data und Data Science Architekturberatung: R on Spark mit SparklyR vs. SparkR, Hive/Beeline Query Optimierung, Integration mit Teradata QueryGrid/Teradata Connector for Hadoop (basierend auf Sqoop).
3. Konzeption/Entwicklung von AbInitio ETL-Pipelines mit GDE/TRMC/EME, Express>It (BRE), Conduct>It (CC), Query>It, Metadata Hub (EME).
4. Vorschlag und Mit-Auswahl von BI & Analytics Use Cases: Promotions (Angebote/Preisveränderungen (PV)), Dynamic Pricing, Backschema, Category Management, Palettenfaktor, Kollisortierung, Shopping Missions, Einkaufs-Planung, Logistik-Planung, Rücksende-/Rückläufer-/Remittenden-Planung.
5. Mitarbeit im Predictive Modelling von Marketing- und Logistik-Prozessen und der Vorhersage des Effektes von Sonderangeboten und diversen Werbemaßnahmen.
6. Beratung zur Auswahl eines Workflow-Management-Tools Oozie, ActiveBatch, Azkaban (LinkedIn), Airflow (Airbnb), Scripting.
7. Berechtigungskonzept mit Apache Ranger, Rechte-Datenbank & LDAP für Hortonworks Hadoop miterstellt.
8. Erstellung von Cross-Platform Packaging-, Versioning-, Deployment- und Dependency-Management-Konzepten für Python, R, Big Data (Spark, Hive, etc.), Teradata, SAP, Ab Initio, MicroStrategy mit Conda/Anaconda, Python, sbt, Java 9 Platform Module System (JPMS) = Project Jigsaw, etc.
9. Virtualisierungskonzepte erstellt für alle Tools mit VMware, Docker, Rancher und Kubernetes, einschließlich Netzwerkkonnektivität, Debugging, Tracing und Monitoring-Funktionen.
10. Erstellung eines 400-seitigen Test-Management-Konzepts incl. ETL- und BI-Testing mit IT-Security für 6 Test-Umgebungen sowie für Python, R, Big Data (Spark, Hive, etc.), Teradata, SAP, Ab Initio, MicroStrategy, Continuous Integration/Deployment mit Jenkins und Sonar(Qube).
DS-Ansatz (Data Science): Random Forest, Gradient Boosting (GBM(Gradient Boosting Machine), XGBoost), CatBoost, LightGBM, SHAP (SHapley Additive exPlanations), stacked ensembles, blending, MART (Multiple Additive Regression Trees), AutoML, Auto-Keras, Dopamine, Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC, Cubist (Erweiterung von Quinlan's M5 model tree), Zeitreihenanalyse, Assoziationsanalyse, (Non-)Linear Regression, Multiple Regression, Anomalie-Erkennung, Apriori-Analyse, Warenkorbanalyse, Überwachte Klassifizierung, Link-Analyse-Netzwerke, Maximum-Likelihood-Schätzer, klassische und mehrstufige Verfahren zur Betrugserkennung (siehe gesonderten Abschnitt), ML-Debugging/Explainable AI im Kontext von LIME, SHAP, partial dependency plots [model leakages, decision explanations in if-statements, …]; Model-Storage in PMML mit angepasstem OpenScoring.io (mit Spring) und Apache Phoenix.
Bibliotheken / Tools Red Hat OpenShift, Docker, Kubernetes, Rancher, R, Big Data (Spark, Hive, Oozie, etc.), Teradata, SAP CAR (Customer Activity Repository 2.0), SAP HANA, SAP BW (Business information Warehouse), SAP BO (Business Objects Business Intelligence), Bex Analyzer, Analysis for Office (AfO), Ab Initio (GDE/TRMC/EME, Express>It (BRE), Conduct>It (CC), Query>It, Metadata Hub (EME)), MicroStrategy, QlikView, MS Visio, Java 9 mit Java Platform Module System (JPMS) = Project Jig-saw, maven, Risk-Based Testing, Apache Ranger, Python: Airflow, Nose2 test suite, Egg packaging, SparkR/SparklyR, webMethods (ESB der Software AG), Scrum, SoS (Scrum of Scrums), LeSS (Large Scale Scrum).

7/2017 - 9/2017

Coach: Big Data Architektur & Data Science (Freiberuflich)

1. Konzeption und Implementierung von Inspectrum, einem Big Data & Apache Spark Data-Flow-Instrumentation & Configuration Framework in Scala: Über JSON/HOCON (Human-Optimized Config Object Notation) Konfigurationsdateien konnten am Ende beliebige Datenflüsse über Spark und sein Ökosystem (incl. Umsystemen) konfiguriert statt programmiert werden mit erheblicher Zeitersparnis. Anbindungen wurden konzipiert für Hive, HBase, Couchbase sowie eine Daten-Filter-Komponente und Virtualisierungen der Komponenten mit Docker, Kubernetes, Rancher.
2. Architekturberatung bzgl. Real-time Use Cases und deren Umsetzung, Datenbanken, Data Science Algorithmen; Architektur von HBase-Datenstrukturen; Pro-Contra-Beratung zum Einsatz von Apache/Cloudera Kudu.
3. Natural Language Processing (NLP): Analyse von Kunden-Feedback/Stimmungen mit spacy.io, Apache OpenNLP (Natural Language Processing), NLTK (Natural Language ToolKit: tagging/chunk parsing), Apache UIMA (Unstructured Information Management architecture/applications).
4. Data Science Beratung: Vorschlag von Verfahren zur Informationsgewinnen fürs Marketing, für Produkt-Analyse und Security-Analysen sowie für den Avira Boot Optimizer. Vorschlag von

Algorithmen für die Nutzung/Analyse der gewonnenen Infos, etwa durch das In-Product-Messaging, den Antivirus, etc.
5. Datenschutz Grundverordnung (EU-DSGVO) / General Data Protection Regulation (EU-GDPR) (Regulation (EU) 2016/679): Beratung zur Legalität der Verbindung von Nutzungs- und Kundendaten und deren Nutzung zu Marketing-Zwecken.
6. Integration von SailPoint IAM mit Big Data über Apache Sentry.

DS-Ansatz (Data Science): Zeitreihenanalyse, Anomalie-Erkennung, Apriori-Analyse, Überwachte Klassifizierung, Gradient Boosting (XGBoost), CatBoost, LightGBM, SHAP (SHapley Additive exPlanations), stacked ensembles, blending, GBM(Gradient Boosting Machine)/MART (Multiple Additive Regression Trees), AutoML, Auto-Keras, Dopamine, Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC, Assoziationsanalyse, Abhängigkeitsanalyse zur Optimierung der Boot-Zeiten, Maximum-Likelihood-Schätzer bzgl. Marketing-Maßnahmen-Effizienz und Konvertierung vom Free-Antivirus-Nutzer zum zahlenden Kunden.
Bibliotheken / Tools OpenShift, Cloudera Hadoop, Apache Spark, Couchbase, HBase, R, Python, SparkR, CentOS, Intellij IDEA, git, Github, Docker, Kubernetes, Rancher, Apache Sentry, Scrum-Prozess.

5/2017 - 8/2017

Coach: Big Data Architektur, Data Science Aspekte sowie Use-Case-Bewertungen (Freiberuflich)

1. Marketing-Strategie Beratung per Design Thinking mit Customer Journey Mapping und Dokumentation der Kunden-Firmen-Touchpoints bzw. Interaktionen, Vermittlung des relevanten Wissens zu den neuesten Programmatic Marketing Ansätzen und den entsprechenden Data Science Grundlagen. Einführung in Customer Data Platforms (CDPs) und Marketing Automation Platforms (MAP). SWAT-Diskussionen (Strengths/Weaknesses/Opportunities/Threats) dazu initiiert und geleitet.
2. Recherche von möglichen Anbietern in obigen Bereichen mit Schwerpunkt auf Customer Intelligence (CI), Customer Data Platforms (CDPs) und Marketing Automation Platforms (MAP) und Kontaktieren der Anbieter: IBM Interact, Oracle Real-Time Decisioning (RTD), SAS Customer Decision Hub, Pega Customer Decision Hub, Adobe Marketing Suite/Cloud, Prudsys, SC-Networks Evalanche, PIA/Dymatrix DynaCampaign, DynaMine, CrossSell, ComArch, FIS Global, DMP-Produkte (AdForm, The Adex, Annalect, Otto, Xaxis Turbine, Acxiom, ...).
3. Erarbeitung der Use-Cases nach Use Case 2.0 Ansatz (inclusive MVP – Minimal Viable Product) mit dem Marketing-Fachteam (besonderer Fokus auf mögliche Echtzeit-Anforderungen/Use Cases) und Bewertung der möglichen Cash Flows sowie der verschiedenen KPIs wie ROI, NPV (Net Present Value), IRR (Internal Rate of Return), WSJF Verspätungskosten (Weighted Shortest Job First), NPS (Net Promoter Score), NBI (Net Banking Income). Anschließende Einführung von weiteren Lean-Startup-Prinzipien sowie Microservices, Evolutionary Architecture, Mobile App Anbindung und passendem Versioning.
4. Datenschutz Grundverordnung (EU-DSGVO) / General Data Protection Regulation (EU-GDPR) (Regulation (EU) 2016/679): Beratung zur Legalität der Verbindung von Nutzungs- und Kundendaten und deren Nutzung zu Marketing-Zwecken.
5. Erstellung einer Baseline-Hadoop-Architektur mit Aufwands-Schätzungen als mögliche Make-Lösung auf Basis von Apache Spark mit Streaming, Alluxio Caching, QBit Microservices, Aerospike DB, Cassandra DB, jBPM, Drools, Oryx 2, WEKA, MOA, Sqoop 1/2, SAS. Diese diente dann auch dem Einkauf zur Preis-Verhandlung.
6. Beratung zu möglichen Data Science Algorithmen rund um das KNIME-System zur Kundensegmentierung und der Ableitung von Produkt- bzw. Marketing-relevanten Affinitäten/möglichen Kundeninteressen und Kundenpfaden: DynaMine, Gradient Boosting (XGBoost), CatBoost, LightGBM, SHAP (SHapley Additive exPlanations), stacked ensembles, blending, GBM(Gradient Boosting Machine)/MART (Multiple Additive Regression Trees), AutoML, Auto-Keras, Dopamine, Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC, Nichtlineare Regression, Random Forests, C4.5, etc.
7. Beratung des Parallelprojekts „Corporate Data Hub" (Digital Transformation / Digital eXperience (DX) Plattform) auf Basis von Spark, Cassandra DB und PostgreSQL, insbesondere bzgl. Anbindungs-Möglichkeiten mit den Marketing-Lösungen und wie diese als PoC (Proof of Concept) für den Data Hub verwendet werden können.
8. Konzeption einer Dynamic Offering Erweiterung HintLog für Dymatrix DynaCampaign: Mit minimalem Aufwand konnten so alle Teilnehmer an Bonus- oder Marketing-Programmen Nachrichten erhalten, wenn irgendwelche Fehler auftauchten oder sie aufgrund von Detail-Regelungen Gefahr liefen, aus dem Programm herauszufallen: Kunden haben dann meist Nachfristen bekommen und so konnte durch das Vermeiden ärgerlicher Situation der NPV-Wert (sprich: die Kundenzufriedenheit) stark gesteigert werden.
9. Review der bestehenden BPM-Modelle in Camunda und Erweiterung dieser Modelle in Camunda um neue Marketing/Kampagnen Use Cases.

10. Konzept erstellt zum semantischen Analysieren und Steuern von Marketing-Kampagnen nach z.B. Kundeninteressen, Kundensituationen, aktuellen Markttendenzen sowie Firmen-Interessen, z.B. als kombinierte/konzertierte Rabattaktionen über verschiedene Teile des Angebots hinweg oder um übergeordnete Marketing-Aussagen in untergeordneten Aktionen immer wieder zu re-iterieren und insgesamt Konsistenz und Stringenz in den Aussagen zu erreichen. Erkannte Kunden-Sitationen/Segmente, Interessen und Unterstützungsbedarf kann so möglichst zielgenau eingesetzt werden, so dass es von den Kunden als hilfreich geschätzt wird und später aus einer Vertrauensbasis heraus (Produkt-/Service-)Empfehlungen gegeben werden können.
11. Natural Language Processing (NLP): Analyse von Kunden-Feedback/Stimmungen mit spacy.io in Python (Net Promoter Score (NPS) Erhebung und Verbesserung).
12. Mitarbeit beim Digital David Projekt als Technologie- und NLP-Berater, der Erstellung eines Chatbots mit IBM Watson Technologie (mittlerweile bei consorsbank.de im Kundenbereich online): Vision: Chatbot der alle Invest- und Banking-Präferenzen der Kunden kennt incl. Konto-, Depot- und WKN-/ISIN-Nummern mit Charts/Trends/Abhängigkeiten und alle Suchen nach Anlagemöglichkeiten durchführt (mit RoboAdvisor im Hintergrund) und daher hohe Kundenbindung und hohe Verkaufskennzahlen erzielt. Meine Arbeit: Analyse der zu erwartenden Text-Dialog-Scripting Aufwände (aufgrund der technisch veralteten Funktionalitäten für Chatbot-Entwickler) und der Total Cost of Ownership (TCO) der IBM-Watson-Lösung und Gegenüberstellung mit einer neuen DLNLP-Architektur (Deep Learning Natural Language Processing) basierend auf Open Source zwecks Preisverhandlungen der Beschaffung: Elemente meiner Open Source Chatbot-Architektur mit DLNLP Tools (Deep Learning Natural Language Processing): Seq2seq, word2vec, ULM-FiT, ELMo, OpenAI Transformer / GPT, Transfer Learning, OpenAI Transformer, spaCy, Stanford CoreNLP, AllenNLP und Virtualisierung mit Docker/Kubernetes zum Training in der Cloud.

DS-Ansatz (Data Science): Zeitreihenanalyse, Anomalie-Erkennung, Apriori-Analyse, Überwachte Klassifizierung, Assoziationsanalyse, Maximum-Likelihood-Schätzer, Kunden-Segmentierungstechniken z.B. nach Personas mit KNIME, DynaMine, Gradient Boosting (XGBoost), CatBoost, LightGBM, SHAP (SHapley Additive exPlanations), stacked ensembles, blending, GBM(Gradient Boosting Machine)/MART (Multiple Additive Regression Trees), AutoML, Auto-Keras, Dopamine, Generalized Linear Models (GLM), Distributed Random Forest (DRF), eXtremely Randomized Tree (XRT), Labeling/Labelling, Bootstrap aggregating (bagging), Receiver Operating Characteristic (ROC)/AUC, Nichtlineare Regression, Random Forests, C4.5.
Bibliotheken / Tools RedHat OpenShift, Red Hat 3scale API Management, IBM Watson, Cloudera, Apache Spark mit Streaming und MLlib, Cassandra DB und PostgreSQL, Aerospike, KNIME, DynaMine, SAS, DynaCampaign, MS Visio, Sparx Enterprise Architect, Camunda, JBoss Drools, Scrum-Prozess, LeSS (Large Scale Scrum.

2/2017 - 5/2017

Chief System und Big Data Architekt (Freiberuflich)

Review und Verbesserung der vorgeschlagenen Grob-Architektur, Ausarbeitung des Architektur-Dokuments auf Basis zahlreicher Meetings und E-Mails mit dem Fachbereich; Vorschlag von Datenmodellen zur redundanzfreien Konvertierung/Speicherung/Aufbereitung und Auswertung aller bestehenden Bank-Transaktionen. Konzepte erstellt für Back-Office-Verarbeitungsverfahren (Reconciliation, Link-Resolution, Transaktions-Bäume/Graphen konzipieren als Struktur und bzgl. Aufbau aus zeitlich versetzt und nur teilweise eintreffenden Informationen) sowie bzgl. komplexer Punkte selbst in Java/Scala entwickelt. Konzeption von Finanz-Planungs-Modellen incl. Steueroptimierung für Wealth Management, Investment-Manager sowie strategische Investitionen. Konzeption der initialen Amazon AWS-Umgebung (benötigt solange die Bank-Umgebung nicht fertig war) und Umsetzung mit AMInator. Anbindung von Apache Sentry an das zentrale IAM-System (Identity & Access Management) der Bank bzw. initial an LDAP. Härtung der Systemkomponenten bzgl. IT-Sicherheit. Konzeption der Spark/Kafka Exactly-Once Verarbeitungsfunktionalität sowie der Gesamt-Business Continuity Funktionalität.

Bibliotheken / Tools
Cloudera Hadoop 5.8 mit HBase + Phoenix, Spark Streaming, MLlib, Alluxio, Kafka mit Camus/Goblin, HDFS, Hive, Flume, Impala, PostgreSQL, Zookeeper, YARN, Hue, Grafana, Cloudera Manager, Apache Sentry, Solr, Splunk + SPL (Search Processing Language), IBM WebSphere MQ, Oracle Weblogic, Sparx Enterprise Architect, Visio, Informatica Data Integration, IBM Integration Bus (IIB) Graphical Data Mapping Editor, JT400/JTOpen, MS Office, Scala, Java, Python.

12/2016 - 1/2017

Architekt/Entwickler Microservices (Freiberuflich)

Zusammentragen der führenden publizierten Techniken und Tools zu Microservices und Mobile Apps & Big Data sowie der integrativen Erstellung von beiden in Form eines ca. 250-seitigen Architektur Blueprints mit folgenden Inhalten: Architekturziele, Architekturprinzipien,

Architekturstandards, Patterns, Neuentwicklung von Konzepten für lokale und vereinfachte Microservices (Neukonzeption eines Code Generierungs-Modells, um viele Microservices in Java/Scala als ein JAR/WAR/EAR oder als mehrere Deployment-Module bauen und debuggen/tracen/testen zu können), Microservice Best Pratices, API Management, Datenkonvertierung/Serialisierung, Logging/Tracing, IT-Sicherheit/IAM, Modellierung per Domain-Driven Design mit Bounded Context, deren Building Blocks und Responsibility Layers, Self Contained Systems (SCS) und Integration der Mobile-App Komponenten, KPI (Key Performance Indicators), Migrationsschritte von Monolithen hin zu Microservices, Software Load-Balancing, Infrastructure as Code, DevOps-Praktiken wie Continuous Integration und Continuous Deployment.
Im praktischen Teil wurde der Code-Generator für die Varianten zur Kombination mehrere Microservices in ein Deployment-Paket oder in je ein Paket entwickelt sowie die unten genannten führenden Bibliotheken für Java, Scala und NodeJS mit AngularJS2 und Ionic Framework (Mobile Apps) getestet.
Bibliotheken / Tools Standard-Tools: Sparx Enterprise Architect, Akka, Apache Gearpump (real-time big data streaming engine over Akka), Apache Flink (actor model, hierarchy, Deathwatch mit libs: CEP, Table, FlinkML, Gelly), spray (HTTP/REST), Spark, HashiCorp Nomad (Clustermanager & Scheduler), SenecaJS, swagger-codegen, Scraml, RAML tools wie JAX-RS Codegen, API Designer, Zipkin/OpenZipkin, OpenTracing, Fluentd (data collector for unified logging), DropWizard, Spring Boot, Spring Cloud (RESTful WebServices in Java), Lagom (Microservices in Scala), Hashicorp-Tools wie Serf, Consul, Nomad (Clustermanager & Scheduler), DevOps- und Continuous Integration/Deployment Tools wie Jenkins, Sonar(Qube), Git, Github, Docker, Kubernetes, Rancher, Chef, Puppet, Axon Framework (Java Microservices), Prometheus (Monitoring), JHipster (yeoman.io, Java & AngularJS microservice generator mit BrowserSync, Spring Boot Devtools [hot reload], Liquibase, Generator for Ionic framework).
Big Data Stack: Thrift, Avro, Spark, Flink, HBase, Cassandra, Hadoop, Cloudera, Hortonworks, Grafana, Hue, vmWare, kvm.
Netflix-Stack: Hysterix (Failure Isolation, Circuit Breaker), Hollow (small to moderately sized in-memory datasets passing from a single producer to many consumers for read-only access), Netflix Conductor (microservices orchestrator), Nebula Gradle plugins, Governator (Guice extensions), Zuul (dynamic routing, monitoring, resiliency, security), Genie (job orchestration), Dyno, Dynomite (storage layer for key-value storage engines), Dyno Queues (Task Queues on Dynomite), Hollow (caching for small read-only in-memory datasets), Astyanax (resilient Cassandra client), EVCache (AWS EC2 memcache), Atlas (In-memory dimensional time series database), Spectator (instrumenting code to record dimensional time series), Vector (performance monitoring framework), Chaos Monkey/Simian Army (failure testing and resilience tools), Spinnaker (continuous delivery platform), Message Security Layer (MSL), Falcor (represent remote data sources as a single domain model via a virtual JSON graph), Restify (node.js REST web service API framework), RxJS (reactive programming library for JavaScript), Aminator (create custom AMIs - Amazon Machine Images), RxNetty (reactive extensions for Netty: asynchronous event-driven network application framework), Ribbon (IPC with software load balancers).
Zalando Mosaic9.org Stack: Tailor (assembling GUI fragments), Skipper (extendable HTTP router for service composition), Shaker (UI components library), Quilt (template/layout storage for Tailor), Innkeeper (RESTful API that stores routes for Skipper).
Med. Standards: ISO 14971, IEC/ISO 62304, Software as a Medical Device (SaMD), European Medical Devices Directive (MDD) 93/42/EEC, EU Medical Device Regulation (MDR 2017/745).

10/2016 - 12/2016 **Big Data Architekt (Freiberuflich)**

1/2016 - 9/2016 **Architekt und Onsite-Offshore-SPOC (Freiberuflich)**

9/2015 - 12/2015 **Technischer Architekt (Freiberuflich)**

7/2015 - 9/2015 **Software Architekt (Freiberuflich)**

5/2015 - 6/2015 **Hadoop Architekt (Freiberuflich)**

12/2014 - 6/2015 **Hadoop Architekt und Projektmanager (Freiberuflich)**

10/2014 - 12/2014 **Sicherheits-Architekt und Entwickler (Freiberuflich)**

9/2014 - 10/2014	**Konzeption JavaScript (Freiberuflich)**
7/2014 - 9/2014	**Konzeption Big Data (Freiberuflich)**
Kontakt:	ARWINET GmbH Tel.: +49 162 7188541 E-Mail: karriere@arwinet.com Homepage:http://www.arwinet.com/