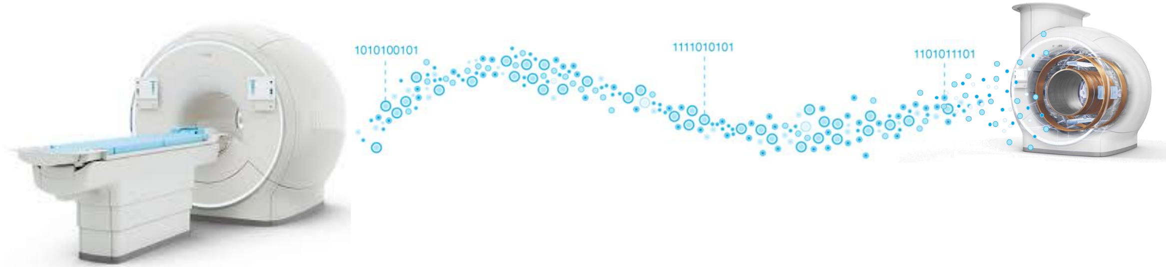
Machine Data for Reference & Platform Architecture evolution



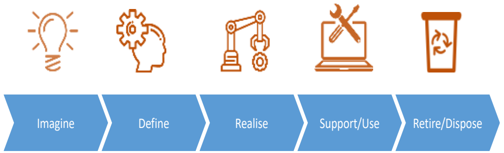
Krelis Blom and Mauro Barbieri
Philips
06 February 2024

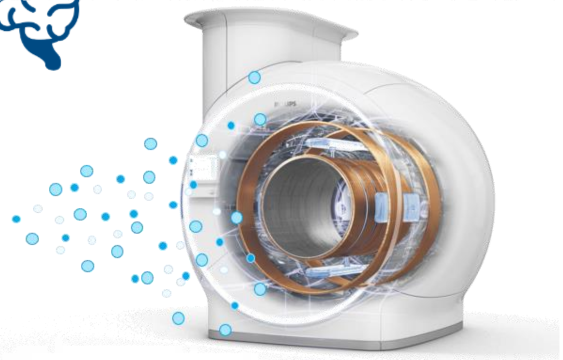
innovation + you

Introduction

To be able to continuously improve, a closed loop is required
Machine data is one data source, next to customer input, knowledge etc.

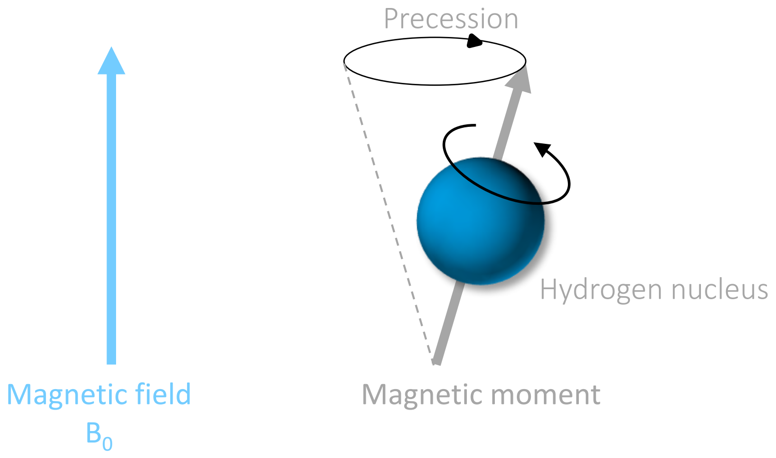
Machine data is used for (predictive) service, reliability improvements, Workflow insights, etc.
Is there a role for Machine Data as input for Reference & Platform Architecture evolution?

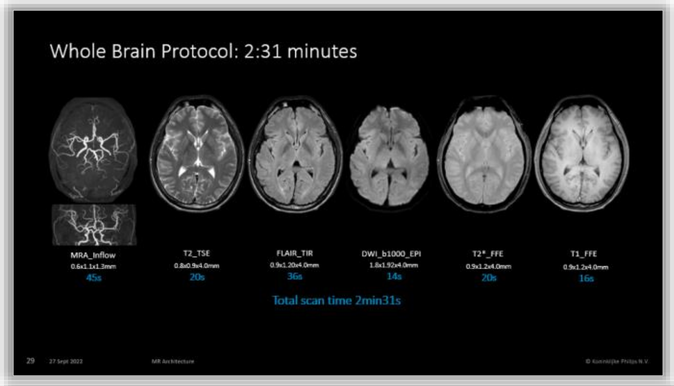




Introduction - MRI

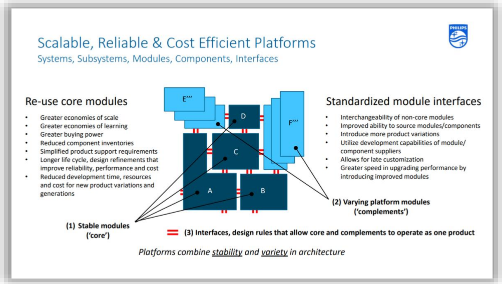
MRIs employ powerful magnets which produce a strong magnetic field that forces protons in the body to align with that field. When a radiofrequency current is then pulsed through the patient, the protons are stimulated, and spin out of equilibrium, straining against the pull of the magnetic field.

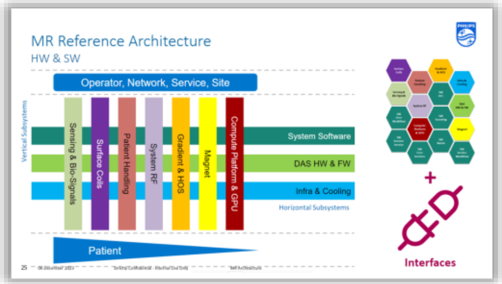




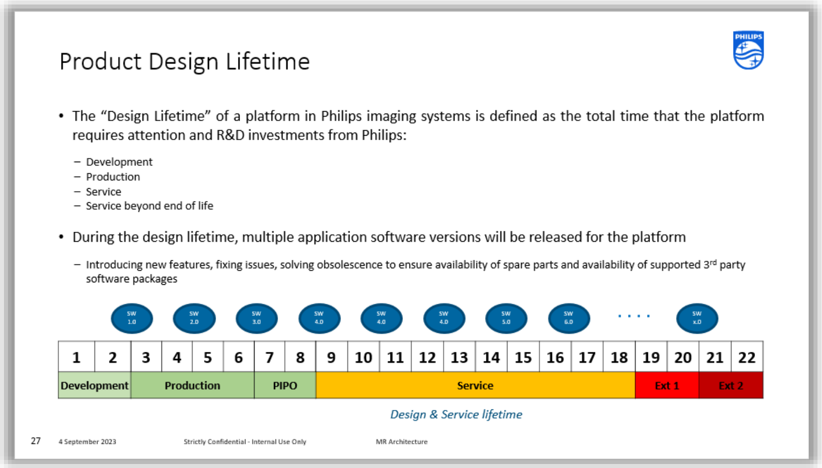
Introduction - MRI

The Reference Architecture aims to define and support governance of a scalable, reliable & cost-efficient MR platform





Introduction - MRI-Lifetime



Challenge:

Repair replacement support for components over a lifetime of 20+ years

Software Upgrade on old compute environments to deploy security, OS, and bugfix updates

Regulatory compliance rules for the health industry, which require up to 1-2 year approval cycles

Data Pipeline challenges

Model Execution
"No data is clean, but most is useful" George Box

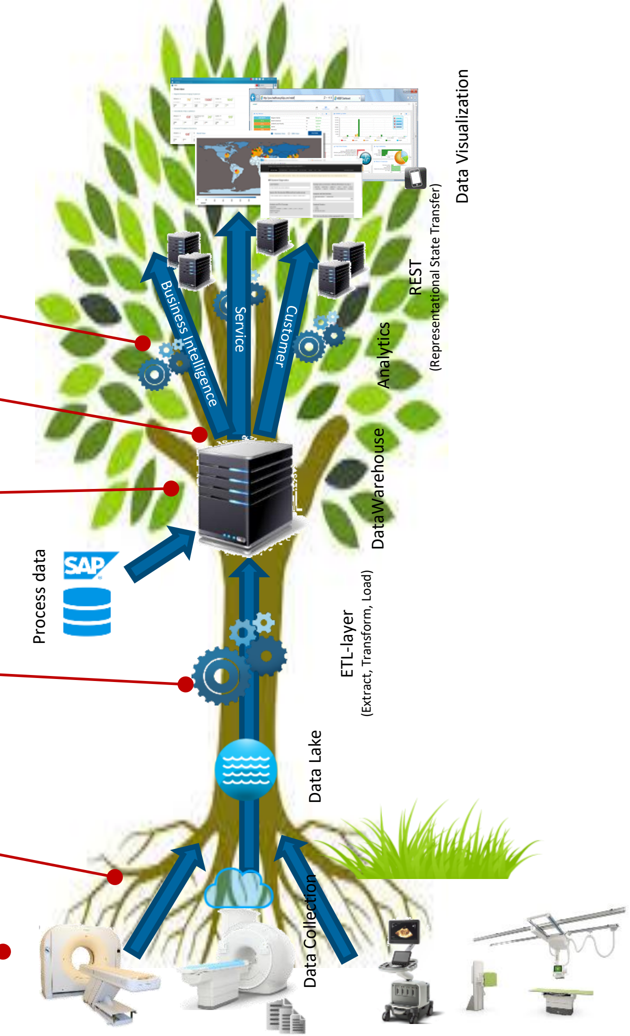
One data structure
"If you torture the data long enough, it will confess." Ronald Coase

Scalability
"We do not let people access the data warehouse – that would slow it down too much"

ETL modularization
"Well, it's all about the ETL law of the transformation of data quantity into data quality"

Availability
"You can't talk about big data without talking about things like privacy and ownership." Rick Smolan

Data Explosion & Data fidelity
"My data sources are unreliable, but their information is fascinating." Ashleigh Brilliant



www.philips.com

Data infrastructure

Mauro Barbieri

innovation you

Challenge of data collection

- Our medical systems generate lots of data during their operations:
  – Event and error logs
  – Sensor data

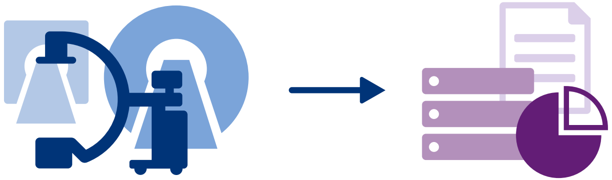
- E.g., every day one MRI scanner logs:
  – 1 million events
  – 200,000 sensor readings
  – Tens of thousands of other data elements

- In many cases, the systems have not been designed with **massive remote** data collection in mind

- Given the lifetime of systems and their nature as *medical devices* we do not always have the freedom to upgrade or redesign for efficient data collection

- Key challenge: **integration of data from existing medical systems**

Our approach

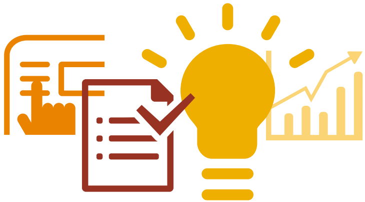


Large collection of historical data available in ONE database

- Event Logs, parameter values, system configuration, etc.
- Failure records / maintenance records / service work orders

For each use case (e.g., product line) one integrated team

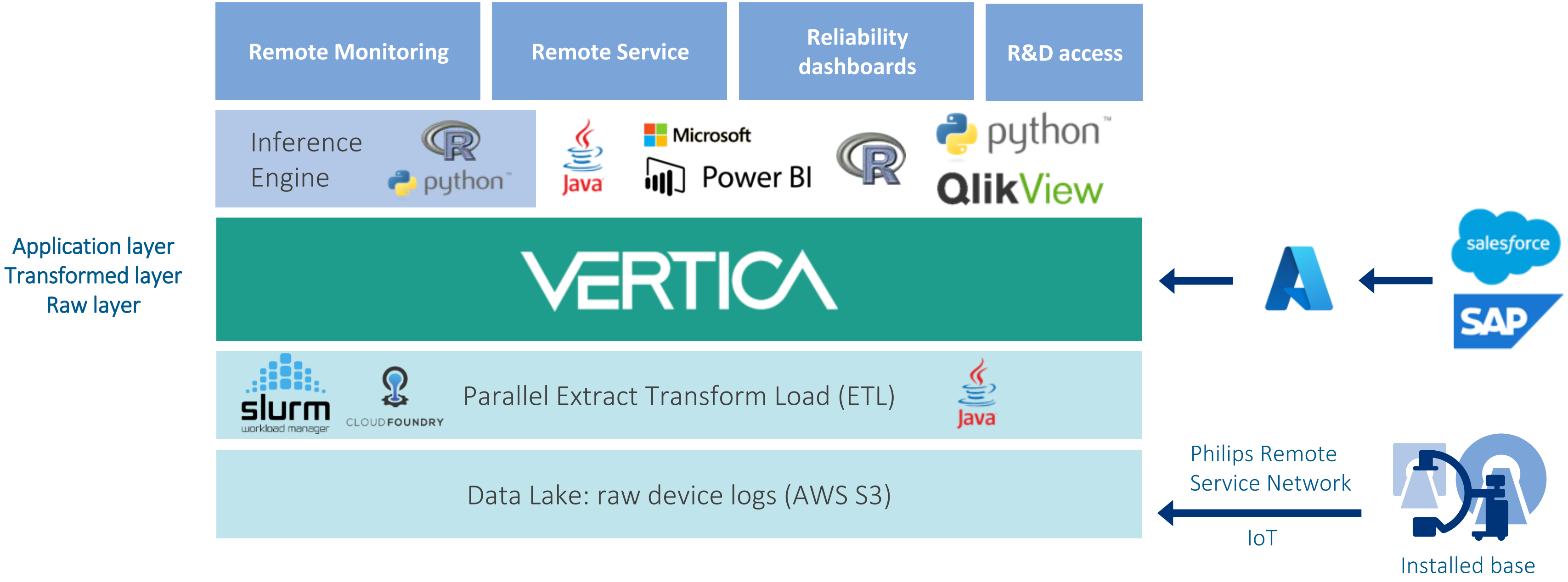
- Business owner from the Service organization
- Data scientists
- Subject matter experts



The team develops

- Insights for specific features, problems, opportunities
- Visualizations of reliability KPIs

High-level architecture



Data integrated

- 200+ ETLs in 30+ data pipelines: connected installed-base MR, IGT-S, IGT-D, IGT-M, DXR, CT, ULS, ICAP) including data from enterprise systems, from factories and repair shops

- 1.6 PB in ~1000 tables and views

- 6 trillion rows of data in one data warehouse
- 5 million queries per day (of which 800k are INSERT/UPDATE/COPY)

- 24/7/365 live data feeds

- 10+ years of historical data

Key aspects

- Simplicity of design: from scratch to production in 8 months (2015)

- Scaling up since 2015 to serve a growing installed-base, larger datasets, more use cases
- In 2022 seamlessly moved from Philips data center in Eindhoven to AWS Cloud
- In 2023 regionalized in China

- Highly stable due to exhaustive error handling
  (millions of medical system log files processed each day)

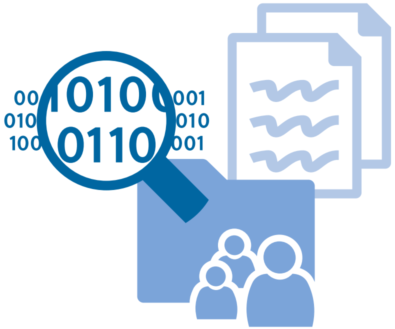
- Live-data pipeline and parallel processing of large historical datasets

- All components have built-in monitoring capabilities
  - Transparency builds trust

Data integration documentation

- **Data model document** specifies how data is extracted and how it is stored in the data warehouse. It includes a description of the source, the transformation and the target (tables/views) as well as the data design and aspects of data quality and governance.
  It includes the failure handling strategy (e.g., import all records or nothing vs. import only valid records) and duplicate avoidance strategy.

- **Data dictionary** explains each column in the data model from an SME perspective (definition and meaning, units of measure, range, expected values, raw value or calculated, etc.).



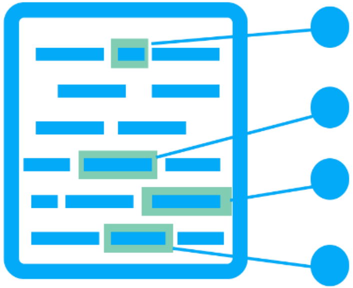
- All documents are reviewed and approved by subject-matter experts, data stakeholders, data scientists and a **data architect**. They are part of our data release process and QMS.

- The documents are available to end-users of the data
- The data warehouse itself as a mini-catalog that links each table/view to its document

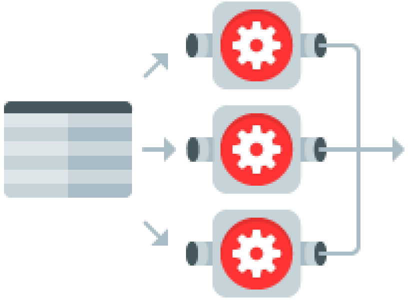
- Data model documents are a cornerstone of DATA **quality**: without data specifications, there can be no verification and no quality assurance

Data transfer, parsing and data lineage

- Scheduled transfers between 5 min to 24 hours depending on the type of device, software version and type of data
- Newer devices support fast transfer of metrics (telemetry)
- On-demand (high-priority) data transfer triggered by Philips Customer Support or by the customer
- Type of device logs: XML, JSON, proprietary text formats (line-based and not), proprietary binary formats, encrypted files, MS windows formats, MS office formats
- We implement ad-hoc parsers or re-use parsers and tools provided by the Philips R&D organizations responsible for product development and service
- Every data point in the data warehouse has lineage information:
  - Where it comes from: medical device identifier as well as detailed location in the file where it was extracted (e.g. byte offset, timestamp, event index, etc.)
  - When it was loaded in the data warehouse
  - Which version of the ETL loaded it

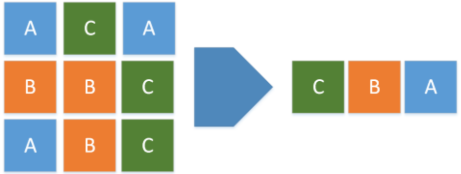


ETL design principles



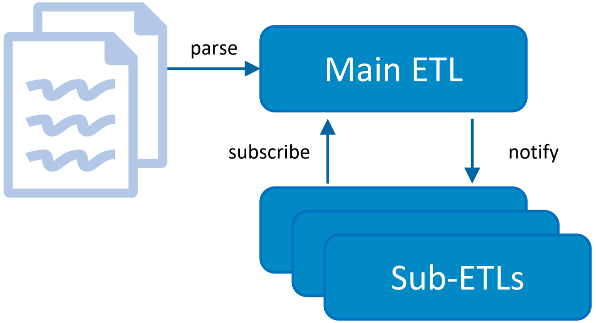
- Independency of ETLs: simplify failure handling and business logic
- Three layers of data
  - RAW
  - Transformed
  - Application

- (Exhaustive) error handling: recoverable vs unrecoverable failures
- Race conditions and re-ordering
- Deduplication strategies



- Historical upload in parallel to live data

Main and Sub-ETLs



- Main ETL
  - Parses and loads data from a certain file type
  - It contains the main logic to parse device files

- Sub-ETL:
  - Cannot run on its own and depends on a main ETL
  - Does not read a file, but it is registered as consumer of data from a main ETL (observer/listener pattern)
  - It may be registered to different main ETLs
  - It may be itself a main ETL

- Example of main ETL: raw events from a main application log (e.g. IGT CDF events)
- Example sub-ETL: extraction of serial numbers, extraction of geo movements, exams and scans (utilization)

PHILIPS

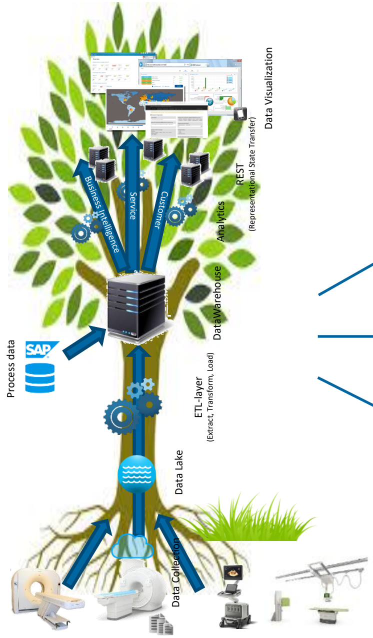
www.philips.com

Impact on Reference & Platform Architecture

Krelis Blom

innovation + you

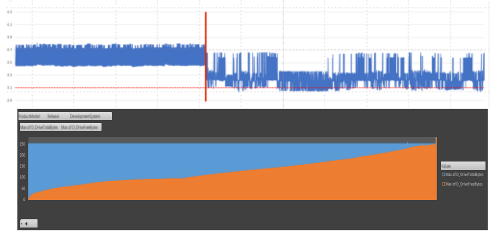
Use of machine data by R&D



Reliability insights
degradation/test-coverage

Utilization insights
usage/permutations

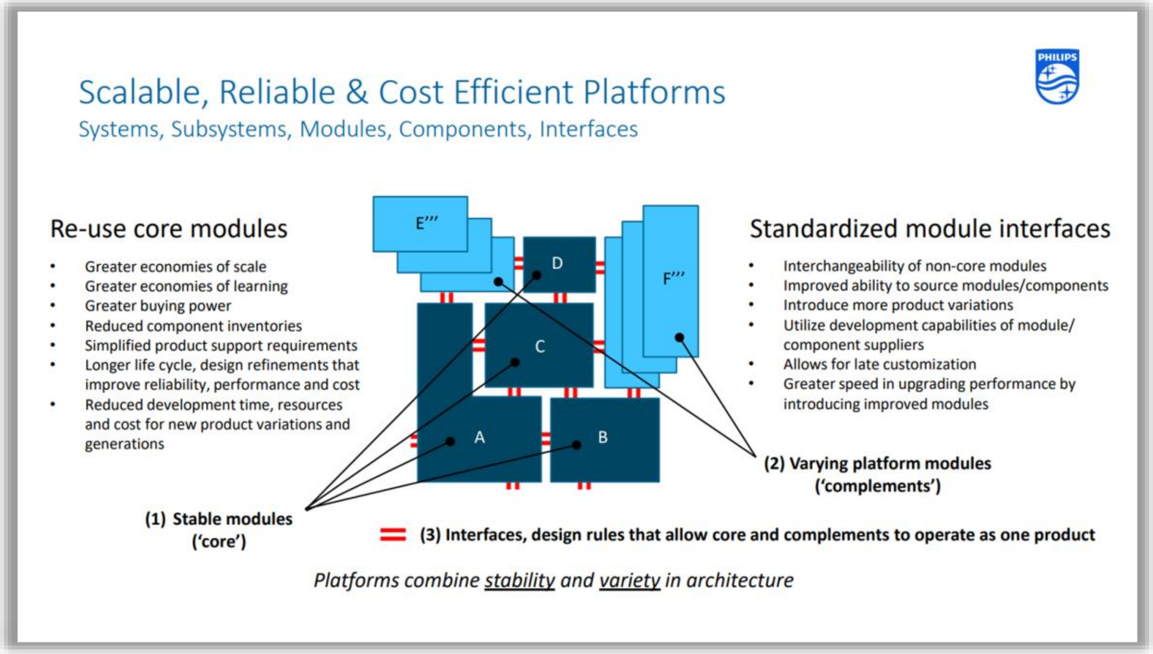
Performance insights
human/environment



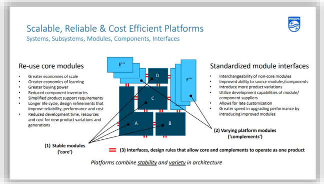
Examples - Platform variants

Supporting multiple products in a cost-efficient way is a balancing act between

Configurable core modules
and
Interchangeable platform components



Examples - Platform variants



Scalable, Reliable & Cost Efficient Platforms

Systems, Subsystems, Modules, Components, Interfaces

Re-use core modules

- Greater economies of scale
- Greater economies of learning
- Greater buying power
- Reduced component inventories
- Simplified product support requirements
- Longer life cycle, design refinements that improve reliability, performance and cost
- Reduced development time, resources and cost for new product variations and generations

Standardized module interfaces

- Interchangeability of non-core modules
- Improved ability to source modules/components
- Introduce more product variations
- Utilize development capabilities of module/ component suppliers
- Allows for late customization
- Greater speed in upgrading performance by introducing improved modules

(1) Stable modules ('core')

(2) Varying platform modules ('complements')

(3) Interfaces, design rules that allow core and complements to operate as one product

Platforms combine stability and variety in architecture

Gradient and cooling components

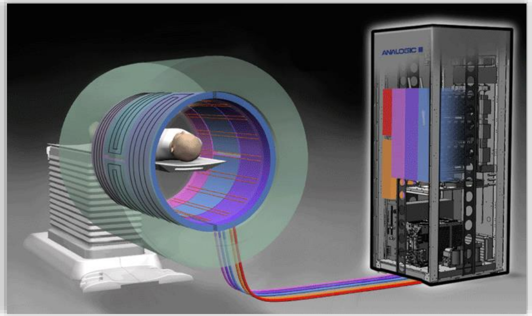
For cooling it is cost efficient to maintain one liquid cooling variant, configurable to support different cooling capacities. (flows)

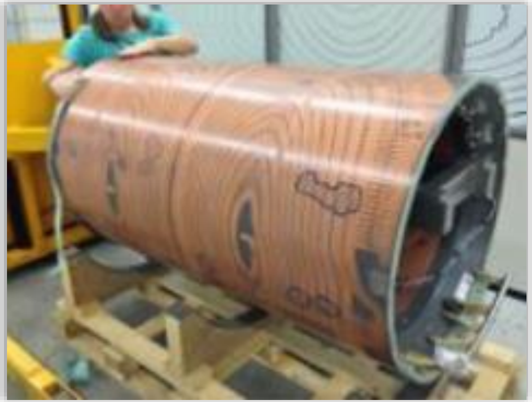
For Gradient power (~ 1 MVA) it is very costly to support all configurations with the most performing Amplifier-Coil combination.

Key question: how many variants are required and where are the switch points?

Modelling & assumption checking with field data is input for decision making

Pitfall: field data is biased. It only shows performance of existing products and features and is weighted with product availability in the market.





Examples – Platform capability checking

Is an update of the system HW required to support image based remote services?

Image based remote services require additional buffer storage of 90GB

Design: 250GB storage available for image data.

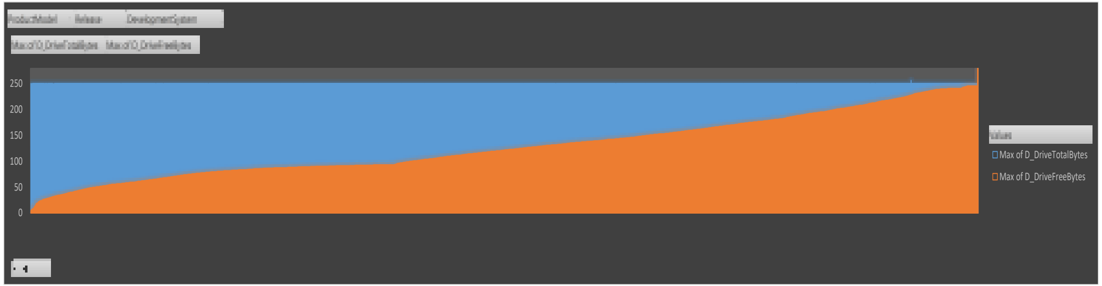
Typical Exam results in 0.5 GB image data / patient
15 exams/day → 7.5GB image data/day

Assumption: enough space available

Key question: is the assumption correct?

Assumption checking: only ~70% of systems in use have enough space available.

Pitfall: Don't torture data.
"Don't focus on raw data saved, those are research sites."
"Outliers should be ignored"
"Only focus on latest systems (cleanup bug solved)"



Summary statement for checking

There is a role for Machine Data as input for Reference & Platform Architecture evolution considering:

- Machine Data is not clean by nature
- Machine Data is biased – it only represents existing products and connected systems
- When “torturing data, it will confess” - a cross-check with model outcomes is required
- Continuous domain knowledge based why questions are required

Machine Data is one of the essential inputs for Reference & Platform Architecture evolution when knowing the weak spots of the data.

The value of machine data is modest in architecture vision creation, more in verifying assumptions and concepts

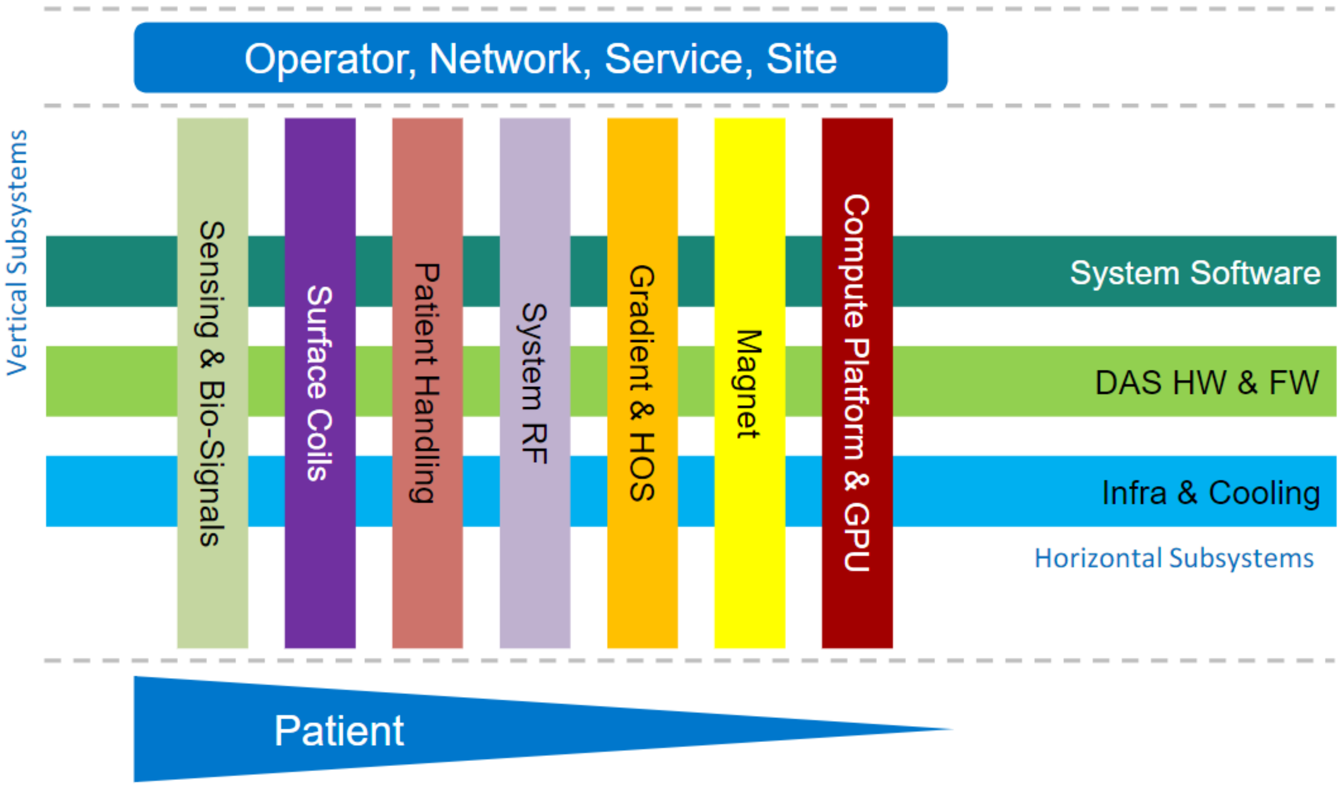
It is easy to lie using data, it is hard to architect without data

PHILIPS

Backup slides

MR Reference Architecture

HW & SW

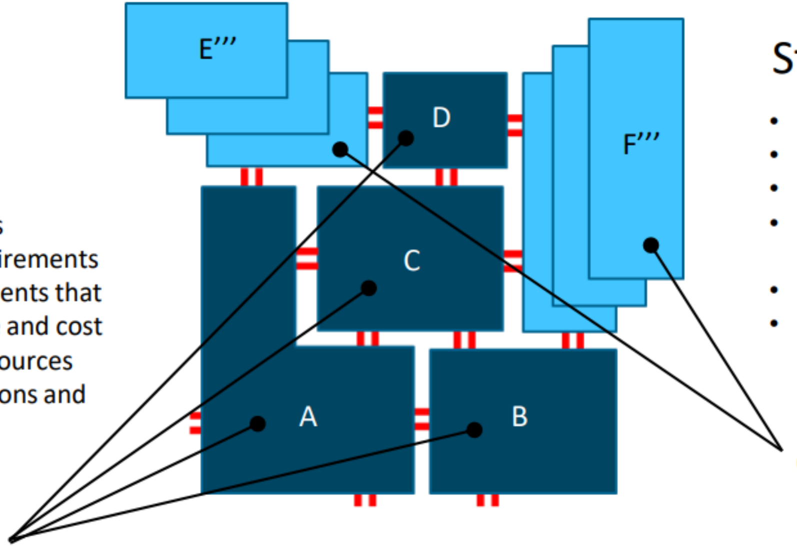


Scalable, Reliable & Cost Efficient Platforms

Systems, Subsystems, Modules, Components, Interfaces

Re-use core modules

- Greater economies of scale
- Greater economies of learning
- Greater buying power
- Reduced component inventories
- Simplified product support requirements
- Longer life cycle, design refinements that improve reliability, performance and cost
- Reduced development time, resources and cost for new product variations and generations



Standardized module interfaces

- Interchangeability of non-core modules
- Improved ability to source modules/components
- Introduce more product variations
- Utilize development capabilities of module/ component suppliers
- Allows for late customization
- Greater speed in upgrading performance by introducing improved modules

(2) Varying platform modules ('complements')

(1) Stable modules ('core')

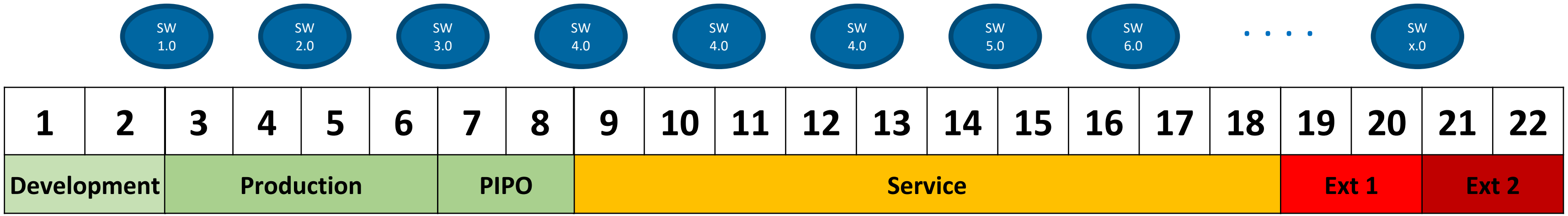
(3) Interfaces, design rules that allow core and complements to operate as one product

Platforms combine <u>stability</u> and <u>variety</u> in architecture

Product Design Lifetime

- The "Design Lifetime" of a platform in Philips imaging systems is defined as the total time that the platform requires attention and R&D investments from Philips:
  - Development
  - Production
  - Service
  - Service beyond end of life

- During the design lifetime, multiple application software versions will be released for the platform
  - Introducing new features, fixing issues, solving obsolescence to ensure availability of spare parts and availability of supported 3rd party software packages



Design & Service lifetime

Summary

Challenges highlighted

One data structure across devices

Challenge: Solutions on top of the data warehouse should be able to use a standardized Interface on high fidelity data for customer facing solutions across devices / business units.
Need: Data dictionary & data cleaning layer on top of Vertica for customer facing solutions specifying *which high fidelity data* is available

Scalability

Challenge: More data is expected as well as bigger compute need for model creation and maintenance.
Need: Policy on cloud data storage and compute resources.

Connected solutions

Challenge: Maintenance of models required a closed loop. E.g.: performance of maintenance models need continuous tracking by inspecting SWO results and alerting incase of performance drifts.
Combining data from multiple sources will bring additional value. (repairshop, Teradata, Sap, ….)
Need: Specifying primary keys enabling data linking between different data sources.

ETL modularization

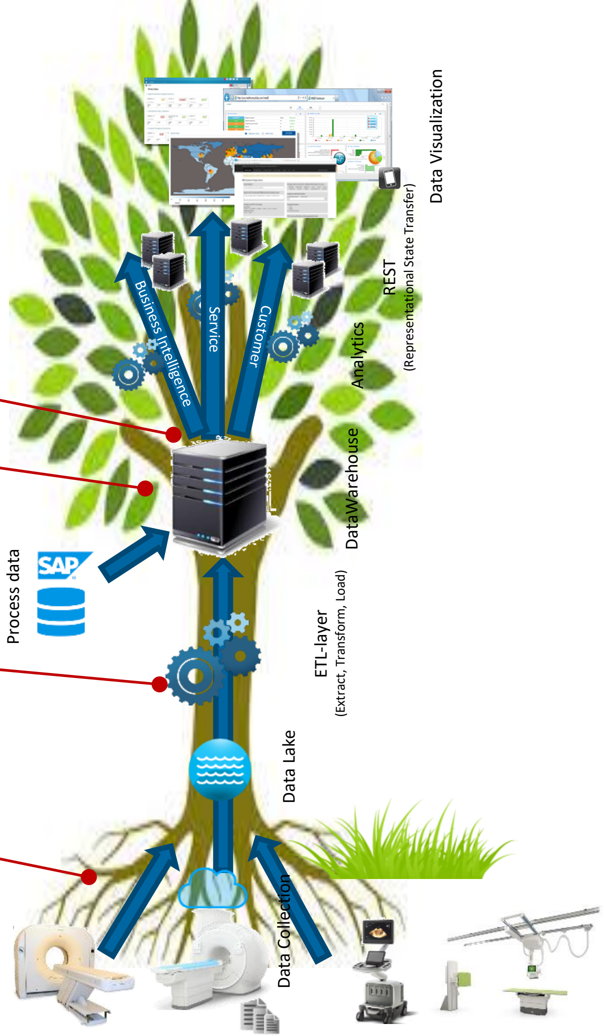
Challenge: For common data like configuration data dedicated parsers are made per device to extract data which all require maintenance.
Need: Design Guidelines/Rules on data format for configuration, testresult and sensor data.
Specifying the *dataformat*

Availability

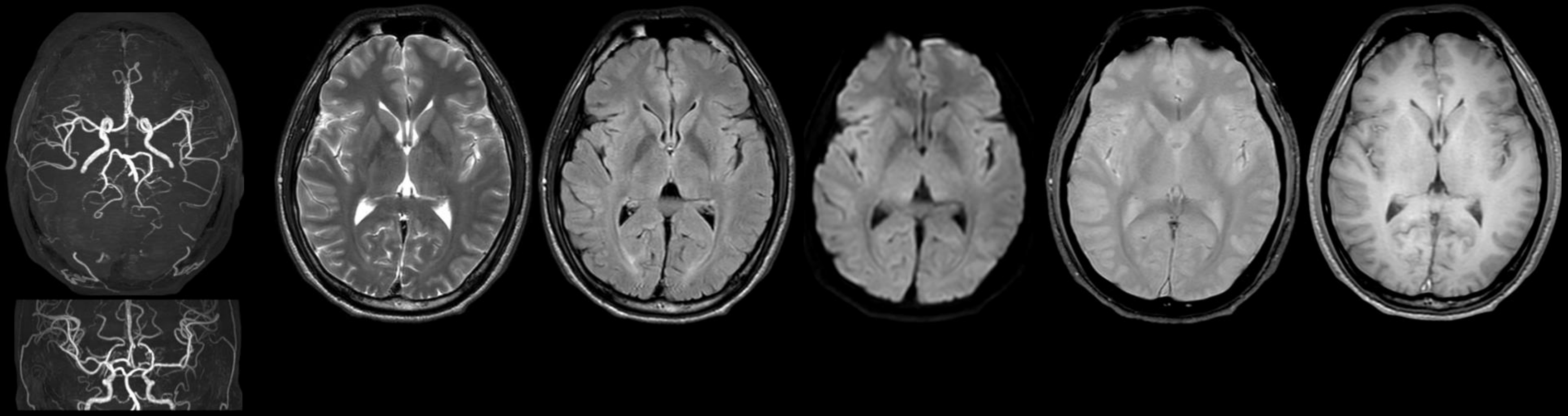
Challenge: Privacy and regulatory rules and concern can lead to systems not being connected or countries not allowing data to be streamed outside premises or borders.
Need: Policy on where to store data and perform analytics. OnDevice, OnPrem or Cloud for all data solutions including predictive maintenance, and customer facing solutions.

Data Explosion

Challenge: telemetric data explodes. This will be a challenge inside the device as well as streaming to the cloud
Need: Design Guideline on how to stream sensor data including latching and filtering

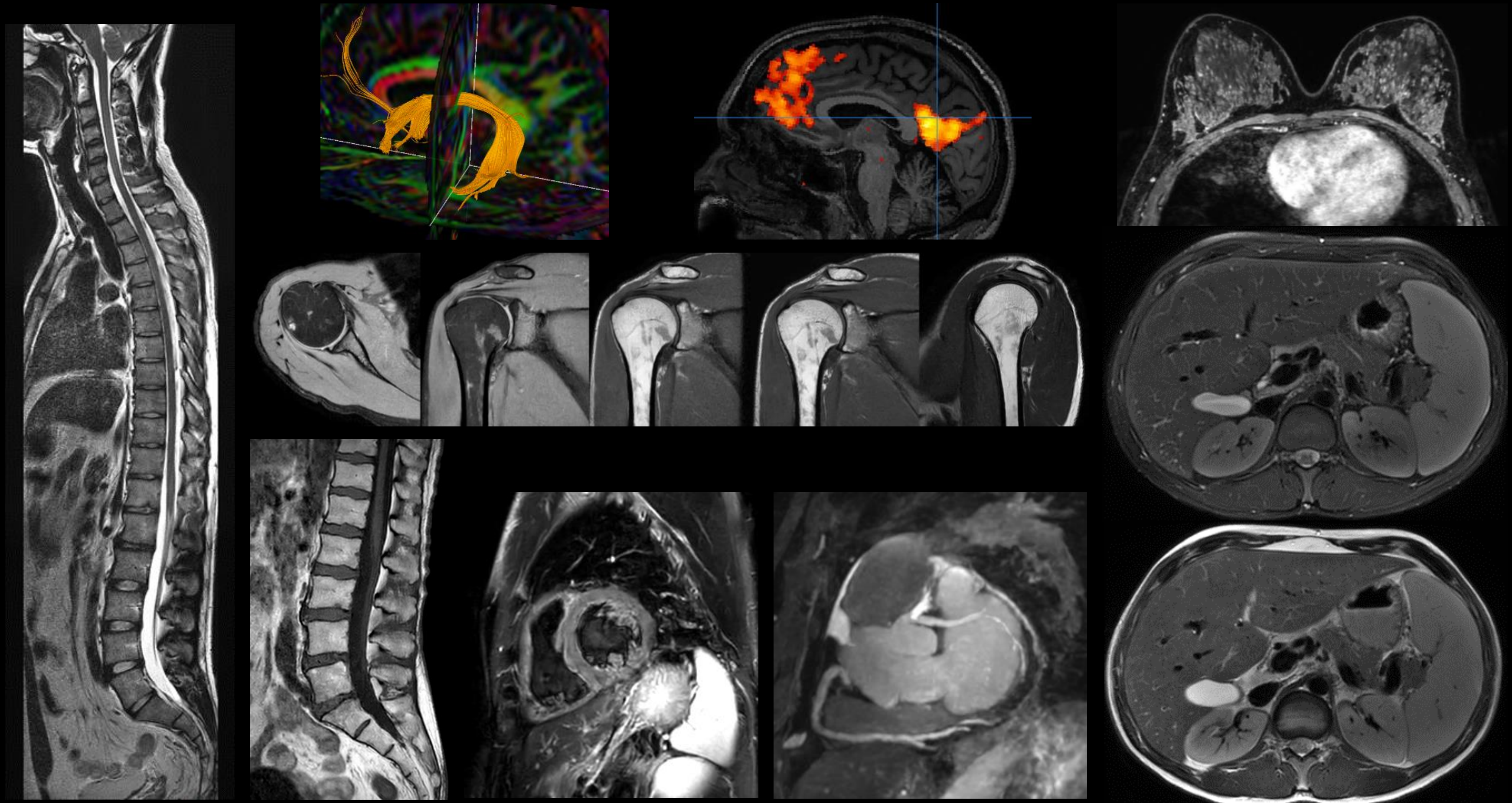


Whole Brain Protocol: 2:31 minutes



MRA_Inflow	T2_TSE	FLAIR_TIR	DWI_b1000_EPI	T2*_FFE	T1_FFE
0.6x1.1x1.3mm	0.8x0.9x4.0mm	0.9x1.20x4.0mm	1.8x1.92x4.0mm	0.9x1.2x4.0mm	0.9x1.2x4.0mm
45s	20s	36s	14s	20s	16s

Total scan time 2min31s



PHILIPS