

# Big Data Tooling and Usage Perspectives in the Airbus Helicopters Test Center

N. Brisset

*Airbus Helicopters, Marignane, France*

[nicolas.brisset@airbus.com](mailto:nicolas.brisset@airbus.com)

## Abstract

In the past years, Airbus Group has been investing into Big Data by developing and setting up infrastructures and backends which have already been presented in various papers. The platform has now reached sufficient maturity to be usable for concrete work and while work is continuing on the backends to improve performance and features, the emphasis is gradually shifting towards end-user tools and operational use cases, among others in the testing domain. New possibilities are appearing, but it also requires some changes in mindsets as commonly used approaches and habits need to be challenged.

This paper will discuss these aspects, focusing on tests performed during development and covering in particular:

- Typical types and volume of data generated during testing
- The need for appropriate data retention policies
- New opportunities brought by big data
- An overview of the palette of tools now available for Airbus Helicopters testers – and SDRs alike – as well as the different use cases it covers
- How some historic data analysis and plotting tools like Sandra can be connected to the big data infrastructure
- Some discussion on challenges and ideas to achieve digital continuity through all test means, from simulation through rigs to flight tests.

**Key words:** big data, digital continuity, timeseries, test configuration, data retention policies.

## Data Generated During Testing

The test center at Airbus Helicopters (AH) is focusing on system-level testing more than software-level testing, but this paper will try to cover both as the boundaries are not always so clear and there are very many commonalities between these two areas anyway.

For vehicle testing the situation is similar: in spite of some quite significant differences in terms of data types and rates, the technical approaches and data strategy seen at a global scale are fairly similar and many challenges are shared with this area.

In terms of data types, during test execution we usually produce a mix of:

- **Raw data** as time series or individual samples, which can vary greatly in terms of frequency, range, amount of parameters or signal types
- **Still images** like for instance screenshots of various displays

- **Videos** (image sequences), either recorded directly at the output of an equipment, or acquired through a camera installed specifically inside the aircraft or sometimes outside the aircraft (e.g. for icing tests)
- **Audio** signals, such as alarms or cockpit conversations.




Based on a subset of all produced data, the tester generates a test result (FAILED or PASSED), which is captured and put into a test report. This test report can be directly produced as proof of compliance (MoC 4: "Lab Test Means"), or provided to a metier specialist who is then in charge of analyzing all test results and producing the final substantiation data (MoC 2: "Analysis").

Note: the amount of data generated by collecting customer data can reach considerable volumes due to the very large number of flights, which compensate for the far smaller number of available parameters per flight. This qualifies

analysis of customer data during operations as very interesting application field for big data technology, but this paper intentionally focuses primarily on data generated during development phases, which have somewhat ironically seen a slower adoption of big data technology at AH.

The table below gives some orders of magnitude of the kind and amount of data generated (not necessarily recorded!) during system testing. When the system contains video signals, particularly ARINC 818, the figures can become so high that it warrants special consideration. Note that the table does not consider the volume that permanent video recording would generate: it assumes only extracts are generated for those parts that can't be analyzed using only still images, and which are run in automatic mode.

*Table 1: Typical Generated Data Volumes during System Testing*

Data type	Generated volume
	~ 5000 to 15000 1 to 10 Go
	~ 100 to 500 10 to 100 Go (only snapshots)
	~ 100 to 1000 < 1 Go
010110 101100 010111	150 A429 15 RS 2 CAN 650 discretes 200 analogues 1 Ethernet network ~ 35 Mb/s (15 Go/h)

### Data Recorded During System Tests

In recent years, AH has not had the means to store systematically all data produced on system rigs and has usually retained only high-level test result evidence: the PASSED / FAILED status. The situation has naturally evolved with the introduction of test automation, which executes automatically during nights and week-ends and produces data that the human operator will need to look at during a post-processing phase. For that to become possible, obviously more data needs to be recorded compared with the

situation where the tester is visualizing the data live during the test.

However, for practical reasons (i.e. the lack of efficient storage means compared with the amount of produced data) the bulk of data collected during these automated test runs is "still", ie a snapshot of a given state during execution but not complete time series. Furthermore, data actually collected usually represents only a small fraction of all that is available.

The test reports are managed in version-controlled environments (typically DOORS) and contain high-level test results, but not the complete associated data. With the advent of big data infrastructures, this is now becoming a viable option, opening new perspectives which we want to discuss in this paper.

### Data Recorded During Vehicle Tests

Vehicle tests (i.e. essentially tests of mechanical parts and dynamic assemblies) are performed in the same department as system tests at AH. Though quite different from avionics system tests due to the nature of the system under test and physical phenomena to observe, there are some interesting facts to share about them. We could sort these tests into four somewhat arbitrary categories using different means:

- *Rotor testing*, which is usually split into different campaigns typically focusing on performance or dynamic behavior
- *Gearbox testing*, which is very specific to these critical parts and for which AH is currently building a whole new rig that will be dedicated to gearbox testing during development phases
- *Iron bird testing*, where we basically put a complete helicopter into a dedicated building and let it run for many hours, either to derisk/reduce test flights or for endurance tests
- *Fatigue, load and environmental testing*, which is a very varied activity where various parts are tested to determine their mechanical characteristics when submitted to environments constraints representative of the most extreme cases that may be encountered during in-service life – and often even beyond to determine margins.

The table below gives some orders of magnitude of the volume and types of data generated during recent campaigns, which are currently stored in data filers under the native recording format, not really suitable for large-scale analysis. For future helicopters, even if we try to contain the increase

of data recording requests, it is clear that at least as much data will be generated, probably even significantly more. Therefore it has to be expected that the current challenges in storing and above all efficiently analyzing these data will only increase.

Table 2: Orders of Magnitude of Vehicle Testing Data Recording Volumes

Test type	Recorded data
Rotor tests	<u>Dynamic tests:</u> 62 runs 45 h 115 sensors Max frequency 5 kHz ⇒ 100 Go ( $\approx 2\text{Go} / \text{h}$ ) <u>Performance</u> 66 runs 37 h 60 sensors Max frequency 5 kHz ⇒ 80 Go data ⇒ 30 Go video
Gearbox tests	50 campaigns / year Max frequency 100 kHz Up to 1 To / campaign ⇒ 300 Go / campaign
Iron bird tests	250 h over 3 years (dev) 350 h in 3 months (endur.) ⇒ 3 To ( $\approx 5\text{Go} / \text{h}$ )
Fatigue, loads & env.	Multiple smaller tests, less standardized. But not dimensioning overall

### Data Retention Policies

Even though in the initial stages of big data introduction many engineers tended to think that there are no limits on storage capacity, nothing comes for free and it is obviously not good practice to accumulate large amounts of data without a clear purpose, and without a clear associated configuration.

Testing during engineering phases, especially when considering the whole palette of possible means from simulation to rigs to real aircraft, can produce very substantial amounts of data. Therefore to avoid filling up the data lake with useless data, following criteria are proposed to be considered for the preliminary definition of retention policies:

- Data recorded during **formal sessions** used to demonstrate compliance should be archived for as long as the tested configuration is in use by customers
- Data pertaining to the demonstration of **safety-critical** functions should be archived for the very long term (e.g. the lifespan of the considered product)
- Data recorded during sessions where an **unexpected event occurred** should be archived until the event is fully understood and resolved, unless any major technical hurdle makes it impossible to leverage the data
- Data recorded during **engineering test sessions** where nothing special happened should be archived until certification is completed
- Finally, in the **absence** of accurate **configuration data** for some archived test results, the interest of keeping time series data should be questioned.

### Expected Benefits of Large-Scale Test Data Recording

There are a number of scenarios in which systematic data storage during development phases can be quite critical to optimize efficiency:

- **Unexpected events** can happen during testing. Sometimes they can't be analyzed on the spot, or even worse: reproduced. In such cases, systematic recording of the data can turn out to be precious to enable analyzing the event after the test session, without the need for repeated test sessions and without the risk of missing out on an important event
- **Comparing with a past configuration** can be very useful when a given test starts failing while it used to work a few weeks or months before. Coming back to the root cause of the difference can be extremely costly or even impossible, and if no data is available for in-depth analysis then the risk is high that the complete root cause remains only partially understood
- **Looking for already performed test points**, be it in simulation, a rig or in flight is expected to bring very significant savings over a full test campaign. It is still too early to give figures, but we do know that sometimes a flight or test for a specific test point is simply redone because it is currently too time consuming to determine whether it has already been done and/or to find the corresponding data. However, in order to bring any real-world benefit the big data

approach must deliver on its performance promise: finding any event through a substantial number of tests shall not take longer than a few minutes! And as highlighted below in the section about challenges, proper configuration data must be associated to the raw data in way that makes it convenient to query

- **Faster data access** even for routine tasks that we perform daily (like plotting curves for test data analysis) can bring very interesting benefits, even though it is more a collateral benefit than the essential reason why we introduce big data. Considering Sandra for instance with a few hundred active users including some people who use it for a few hours every day, the productivity boost they will gain from almost instantaneous access to the data they need can scale up to quite significant savings...

### Big Data Tool Palette

We have seen that data volumes generated during testing phases increase constantly, and that big data will largely determine our ability and efficiency in coping with this huge amount of data. But for that we will clearly need some specific tools, and not only for data storage.

Big data is a rather simple concept that turns out to be very complex in terms of implementation,

and it often requires specific non-trivial tools, adding to the complexity of already demanding test environments. This complexity is impacting users and especially difficult to manage, as the technology is still evolving at a very fast pace and tools are far from mature in this field.

In this context the introduction of new tools must be considered with caution, balancing the benefits of integrating already existing tools with big data technology against the introduction of new specialized tools. The following picture illustrates various types of tools needed for efficient big data usage. They can be broken up into various categories:

- **Data ingestion** tools that allow efficient data pre-processing and ingestion
- **Storage and processing infrastructure** ("data lake") that holds the data in an efficient way, making it possible to retrieve it in a massively parallel and very quick way
- **APIs** that allow accessing the data from a large variety of tools and languages
- **Specialized/metier tools** that are used to develop and run complex dedicated algorithms
- **General-purpose data analysis and plotting tools** that can be tuned either for time series, or for statistical analysis of discrete events

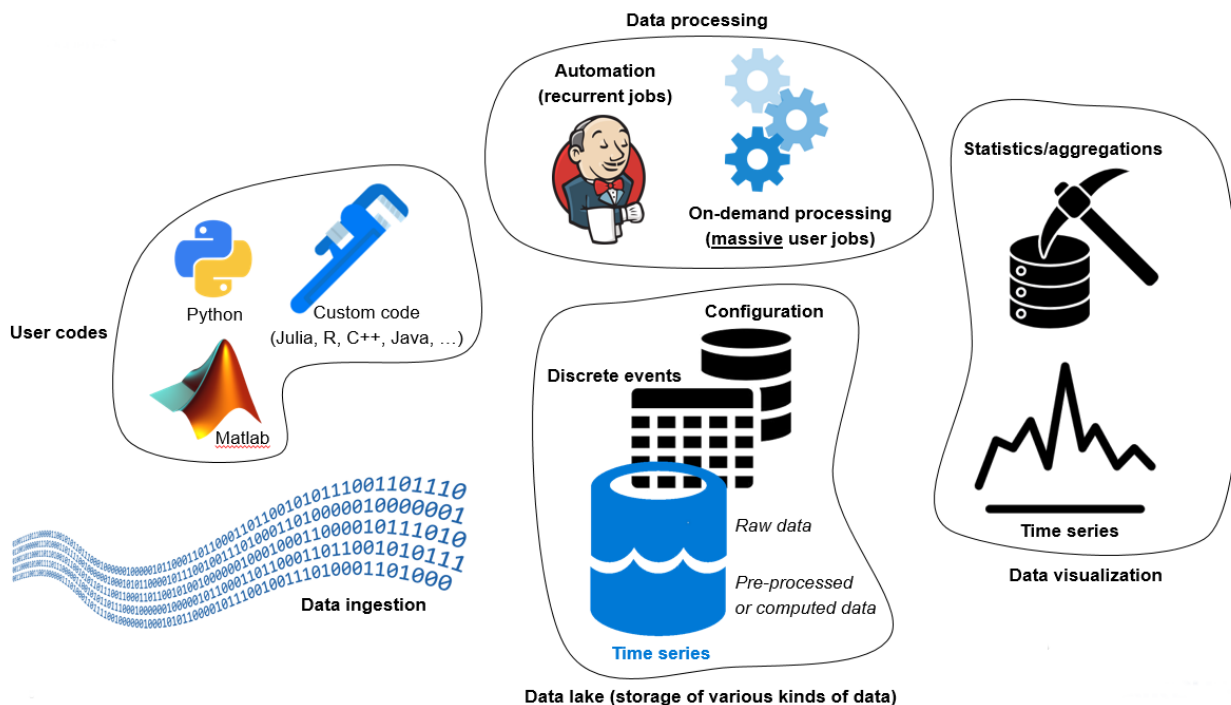


Figure 1: Big Data Tool Palette (Generic)

## Current Tooling Landscape at AH

Big data is still in its early stages at AH, especially when it comes to development test results. There is an extensive on-premises platform which is used for a number of use cases but not yet for the storage and processing of time series. The various components described in a generic way in the previous paragraph are currently supported as follows:

- **Data ingestion** is performed in multiple steps: data is transferred either as CSV (fleet data) or in a HDF5 container (prototype test data), then turned into AVRO buffers and ingested into HBase
- **Storage** components are offered by the TSAS infrastructure, which provides the "data lake" component in the form of a **timeseries** database working together with another database dedicated to the storage of other kinds of data like **events**
- **Massive processing** can be achieved thanks to [TSAS.processing](#), a spark-based engine with a user-friendly web interface for the configuration of search jobs
- **Job automation** can be setup in order to launch data tagging (e.g. Flight Regime Recognition) or data check routines (e.g. CheckFTI)
- **Parameter management and queries** are done thanks to the combination of a specific database, along with a dedicated [TSAS.search](#) web front-end
- **REST APIs** are at the core of TSAS and allow accessing the data from a large variety of tools and languages: Matlab, python (PyTSAS), curl, etc...
- **Specialized/metier tools** that are used to develop and run complex dedicated algorithms
- **Time series analysis and visualization** can be done using either [TSAS.plot](#) (in the browser, stateless no-install usage) or using the well-established Sandra tool, that requires an installation but offers many more capabilities and ubiquity (same tool for all sources)
- **Data statistics, aggregations and event analysis** are preferably performed using Tibco Spotfire, which is the standard tool selected for that kind of task. It can be connected to the event database and leverage all pre-processing done using the TSAS ecosystem
- A generic platform for the user-triggered deployment of **user web apps** built using Dash

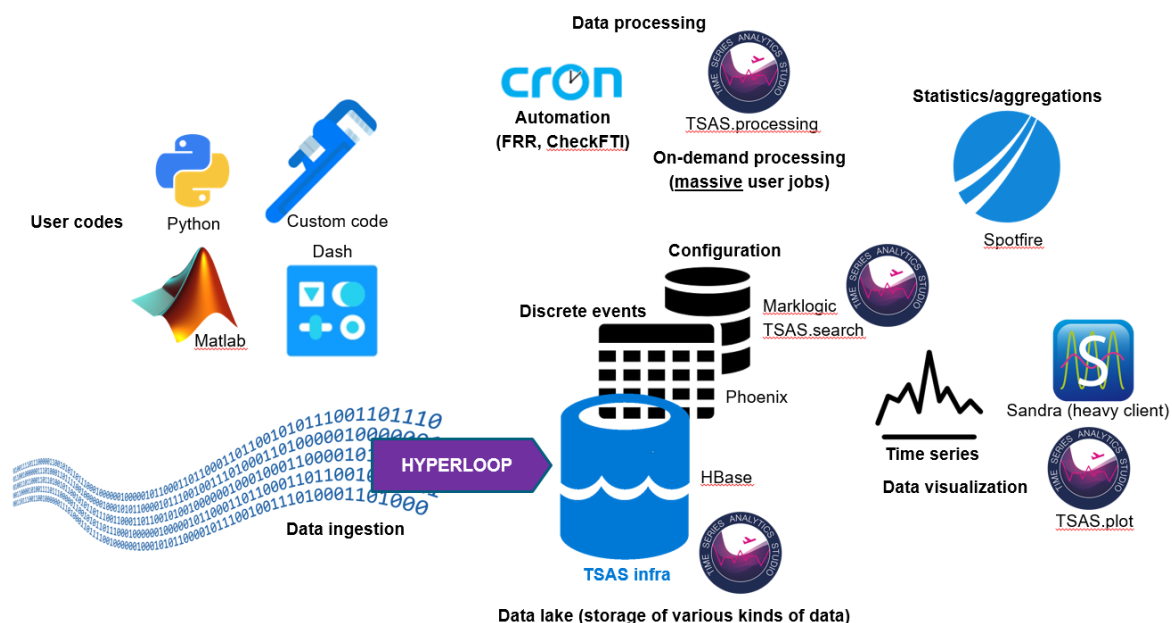


Figure 2: Current Big Data Tools at AH

## Timeline

Big data experiments are already quite numerous at AH, but we are nonetheless still in the early stages. There are currently more use cases developed on data types other than time

series and with data from other sources than development tests (i.e. the focus is rather on customer data). But it will undoubtedly come as the company digitalizes itself and strives for increased efficiency.



Without taking the risk of giving dates that are still quite uncertain, we can estimate that it will take years until big data is fully integrated into daily routine for all areas of the company. Many plans will probably need to be revisited along the

way, but we can try to outline the natural steps through which we will have to go until full production deployment. This is what the next picture attempts to do.

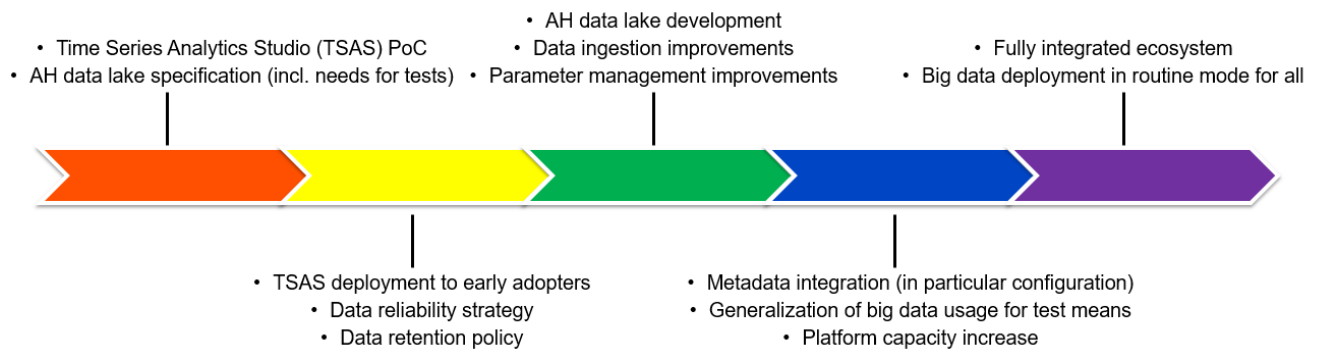


Figure 3: Big Data for Tests: Deployment Outline

### Challenges, Present and Future

A number of unique challenges will come from the massive introduction of big data into testing routine. They are listed below, with some hints as to how they could be addressed in a journey that is only starting.

- **Data security and compliance** will be an essential element, probably one of the main driving factors in terms of technological choices as addressed in previous papers like [1]. The constraints in terms of data protection are likely to force some major big data editors to develop extensive on-premises solutions, or to force companies like Airbus to invest into cloud technologies which are not necessarily their primary area of expertise
- **Configuration data** must absolutely be considered as a very high priority. As challenging as collecting huge amounts of timeseries may be, there are solutions to this problem. But maintaining a consistent, up-to-date, accessible, relevant and reliable configuration index associated with the raw data may prove to be even more of a challenge considering the complexity of current products and industrial processes undergoing a major digitalization effort!
- **Data reliability** is likely to be one of the hardest parts in this endeavour. When people look at data directly, it is realistic to assume that any error in the data will get noticed and be corrected. But what will happen when we start running massively parallelized and unsupervised queries to analyze the data for us, in order to define potentially safety-critical part sizing or maintenance intervals? Tomorrow's design office engineers will have direct and efficient access to enormous amounts of data on his/her own, but how can we ensure that it cannot be misleading?
- **Digital continuity across means** will play an increasingly important role when our new efficiency targets command that our testing fully leverages a large palette of testing means ranging from simulation to real aircraft over all kinds of rigs. Not only will configuration data be key, but data continuity will be adding a substantial amount of complexity to the approach. If a given parameter or data cannot be correlated between the different means, a large part of the benefit (and the ability to rely more on simulation in the future) will be lost
- **Easy to use & learn tools** will be a prerequisite to reap the full benefits of the introduction of big data. Preliminary experience shows that while a small number of highly-skilled, motivated and IT-oriented engineers might be ready to learn a new programming language or a complex new tool, it is by far not given that the majority will find time and/or motivation to do that. It will therefore be of paramount importance to develop easy-to-use and well-integrated tools for the end users, carefully designed to fit their workflows
- **Management of parameter lifecycle** during the whole service life of our aircraft is already a challenge, but it will become even more difficult as we collect and keep data over long periods of time. Flight Test Installations are fairly complex, some sensors age and must be recalibrated, requests change permanently thereby calling for adjustments in the installations. This makes digital continuity even more

challenging, but also more necessary than ever as we will likely be using more and more data thanks to the new possibilities

- **Data volumes** (especially with the development of automatic testing) will drastically increase. The ability to store large amounts of data will likely turn into a form of obligation to store data. The seemingly huge datalakes of today will probably run full much faster than expected, and what happens next is easy to imagine: even larger storage volumes, soaring costs and substantial difficulties in maintaining data consistency and integrity over time. A dangerous factor is also that the cloud is something intangible (unlike manipulating physical media), blurring the perception of actual data volumes being stored. Data acquisition and retention policies will certainly need to be developed, but pruning and cleaning up petabytes of accumulated data will never be an easy task! Not to add the fact that storage formats will probably evolve, and some (typically proprietary formats) may prove to be a deadly trap
- **Obsolescence** will strike, as in any other field of technology. Or possibly even faster considering the relative immaturity of big data technological bricks, from hardware all the way to software components. We have to make choices today to advance on the path that unlocks the most interesting use cases. But having to run before we can walk and developing using agile and incremental approaches makes us prone to errors and the need to rework and refactor. It is therefore critical to carefully lay down the foundations of our future environments! One particularly acute issue will be data formats, because it is likely that these will continue evolving.
- Therefore **open-source or at least fully documented formats must be preferred** to proprietary formats for sustainability
- **Video** is a topic that calls for specific attention because although any modern smartphone can record in 4K at 60 fps or even more, leading to people expecting any video in the professional world to be recorded in 4K at least, this kind of data can represent very massive volumes that take their toll on any infrastructure in terms of processing power, storage size, etc... On top of that, videos can be compressed in many different ways and will most of the time require dedicated tools for their in-depth (automated) analysis, tools that may be sensitive to compression and formats
- Finally, the **mindsets** of engineers in our companies must evolve. Some may believe

that the young generation of digital natives is better prepared and that their increasing presence over the coming years will ease the transition. It may be true, but others may argue that having grown up in a world of unlimited storage, bandwidth and computing power may push them down a dangerous path. We will need to pay attention to training and support for all the new users, and make them aware of the most important challenges to guarantee efficient usage and sustainability of big data in the testing area.

## Conclusion

Even though the previous chapter on challenges may sound scary, it is not the intention of this paper to paint a grim picture of the future of testing in a big data enabled world!

On the contrary, the advent of big data in our development environments opens up very interesting perspectives in terms of being able to detect previously unseen phenomena, catching rare events and generally optimizing testing during development phases. It will most certainly be a pillar of our future efficiency.

Before jumping head first into the data lake at the risk of drowning in it, it is crucial to address all the listed challenges and to ensure that we take the most promising and future-proof paths. And even doing so, we must be ready to adapt as we progress along this exciting and fast-changing path.

## References

- [1] L. Peltiers, Data storage landscape: which solution for which purpose ?, *ETTC'21*, Proceedings p. 101
- [2] A. Miguel Arevalo Nogales, Embracing new generation data access and processing methodologies in Flight Test, *ETTC'21*, Proceedings p. 123