// ARTICLE · PUBLISHED 3 NOVEMBER 2026

DICOM and PACS on MAX4: Pictures Before Patients

A first imaging trial on synthetic scans found two defaults MARA's design has to change before she ever sees a real one

Jason Newell · MAX Research Collective · Article ART-HLC-021 · Published 3 November 2026
Licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), https://creativecommons.org/licenses/by/4.0/
Original: https://www.maxresearchcollective.com/articles/dicom-and-pacs-on-max4.html
Cite as: Newell, J. (2026). DICOM and PACS on MAX4: Pictures Before Patients. MAX Research Collective, ART-HLC-021. https://www.maxresearchcollective.com/articles/dicom-and-pacs-on-max4.html

What the trial found

Medical images are a large part of a patient's record, and MARA, the Medical Agentic Resident Assistant the Collective is designing, will need to read them. Before she does, the plumbing has to be proved somewhere no patient can be hurt. That place is MAX4, the desktop rebuild of MAX, our AI resident, which holds no patient data.

The first trial used two open-source tools, an imaging server called Orthanc and a Python library called pydicom, with three invented patients and 24 synthetic CT slices. It checked who the server refuses, how studies arrive, how they're found and what anonymization removes. Two results shape MARA's design more than the rest, and both are defaults. Anonymizing a study in Orthanc makes a new copy and keeps the original. And it cleans the file's labels while leaving its pixels alone, so text burned into an image survives. Nothing here used real patient data, and nothing here is a medical device.

Why imaging, and why on MAX4 first

MARA's design has her reading the sources that make up one patient's care: the electronic health record, labs, pharmacy, claims and imaging. Imaging is the odd one out. The record, the labs and the claims arrive as structured text through FHIR (the HL7 standard for exchanging health records) or X12 (the standard for claims). Images arrive as DICOM files, through a different family of protocols, from scanners and from archives called PACS. A MARA that can't read them misses a large part of what a clinician looks at.

MARA is built after MAX, on what MAX has already proved. Her requirements say MAX must first prove the substrate: allowlisted reach, a tamper-evident audit, honest grading and published failures. MAX4 is where that proof happens, and its clinical rule is short: no patient records. So the imaging trial runs on MAX4 with synthetic data only.

The trial finds out, with no one at risk, what the imaging plumbing does by default, and which of those defaults MARA's design has to change.

DICOM and PACS in brief

DICOM (Digital Imaging and Communications in Medicine) is two things. It's a file format: a header of tagged attributes (who the patient is, which study, which scanner, how the pixels are laid out) followed by the pixel data. And it's a set of network protocols for moving those files around. The older protocol family, usually called DIMSE, works between named endpoints called application entities, each with an AE title. A scanner sends a C-ECHO to check it can reach the archive, then a C-STORE for each image; other services find (C-FIND) and move (C-MOVE) studies. The newer family, DICOMweb, does the same over HTTP: QIDO-RS to search, WADO-RS to retrieve, STOW-RS to store.

A PACS (picture archiving and communication system) is the archive at the centre. It receives images from scanners, stores them and serves them to viewers and other systems. In a hospital it's usually a large commercial product. In a trial it can be much smaller.

The tools

Orthanc is an open-source DICOM server from Belgium, first developed at the University Hospital of Liège and now hosted by UCLouvain. It speaks DIMSE and, through a plugin, DICOMweb, and it has its own REST interface for everything else. A 2026 description of the project lists its uses as a lightweight PACS, middleware for exchanging images between sites, a substrate for teleradiology, and a platform for de-identification and AI pipelines. The trial ran version 1.13.0, released in August 2026, from the project's Docker image.

pydicom is a Python library that reads DICOM files, changes them and writes them back. The trial used version 3.0.2, released in March 2026, and the version matters: it fixes CVE-2026-32711, a path-traversal flaw in how earlier versions handled DICOMDIR file sets, rated 7.8 (High) by some trackers. Every version from 2.0.0-rc.1 to 3.0.1 is affected. That's why the trial pins version 3.0.2. Its sister library, pynetdicom 3.0.4, speaks the DIMSE protocols.

Both fit MAX's hard rules. They run on the machine, with no cloud call, and they're open source. The licences differ and need care, though. pydicom and pynetdicom are MIT. Orthanc's core is GPLv3 or later, and its DICOMweb and Python plugins are AGPL, which its own README calls more restrictive than the core's licence. The trial runs Orthanc as a separate program in its own container and talks to it only over the network, so no MAX code links to it. What the AGPL means for a MARA that clinics would run is a question for counsel before any deployment. This trial leaves it open.

The trial

The trial is two scripts (Figure 1).

The first, make_synthetic.py, builds three CT studies of eight slices each, for three invented patients named Testpatient^Ada, ^Boris and ^Cleo. Each slice carries the attributes a real study would: name, patient ID, birth date, an address, another patient ID, an accession number, a referring physician, an institution, a study date and a description that repeats the patient's name. Each also carries a private tag (the kind vendors use for anything at all) holding the patient ID again. The pixels are

a generated pattern with one bright rectangle standing in for text burned into the image, and the file says so: BurnedInAnnotation = YES.

Orthanc ran with its ports bound to the loopback address only, login required, and the DICOM listener closed to everyone except one registered AE title, MAX4. Orthanc's own security guide asks for these settings. It says the server assumes it runs "on the localhost, within a secured environment".

The second script, trial.py, then did five things and recorded each:

1. It tried the server as a stranger: a REST call with no login, and a DICOM association from the unregistered AE title STRANGER.
2. It sent one study by C-STORE from the registered AE title MAX4.
3. It uploaded the other two studies over REST.
4. It searched for them over DICOMweb (QIDO-RS).
5. It anonymized every study through Orthanc's REST interface, downloaded every anonymized instance, and checked each one with pydicom against the original.

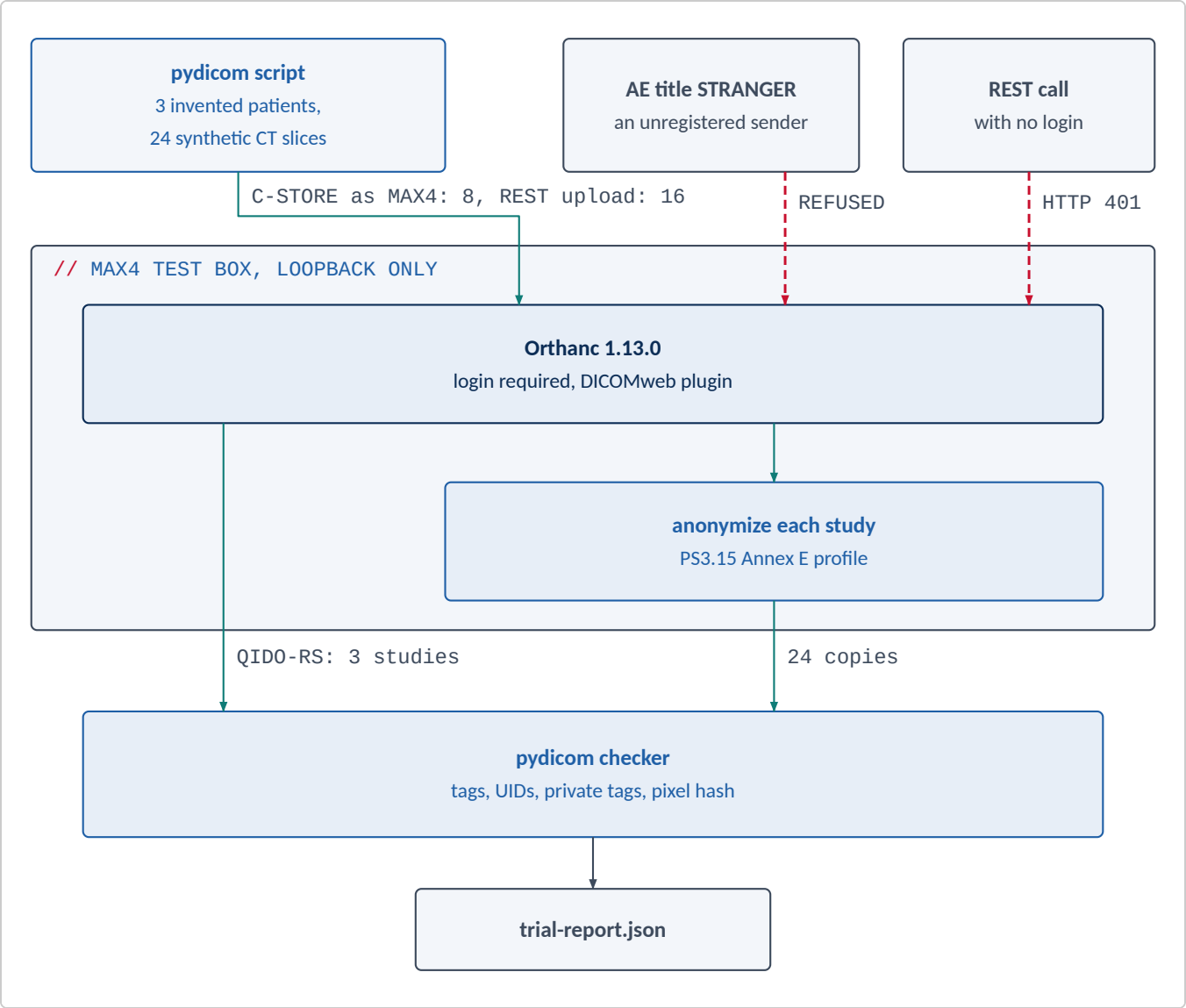


Figure 1. Read down the page: synthetic studies go in by both protocols, strangers are refused, and every anonymized copy is checked against its original.

What happened

Table 1. What each step of the trial returned.

STEP	RESULT
REST call with no login	Refused (HTTP 401)
DICOM association from STRANGER	Refused; nothing stored
C-ECHO and C-STORE from MAX4	Echo succeeded; 8 of 8 slices stored (0.40 s)
REST upload	16 of 16 slices stored (0.05 s)
QIDO-RS search for Testpatient*	3 studies found
Anonymize 3 studies	24 anonymized instances, checked in 0.14 s in total
Identifying attributes left with their original values	None of the 11 checked
Study, series and instance IDs (UIDs) unchanged	0 of 72
Private tags left	0
Pixel data unchanged	24 of 24
BurnedInAnnotation after anonymization	Still YES
Studies held by Orthanc afterwards	6 (3 originals, 3 copies)

The full output is in trial-report.json. The timings come from a cloud test container with tiny images, so they show that nothing was slow and say little about how fast a clinic's server would be.

Orthanc did what its documentation says. By default it removes the attributes listed in Table E.1-1 of the DICOM standard's confidentiality profile (PS3.15, Annex E; this release defaults to the 2023b edition, while the current edition is 2026d), drops private tags, replaces the UIDs, and records what it did: each anonymized file says PatientIdentityRemoved = YES and DeidentificationMethod = Orthanc 1.13.0. The

patient ID became a random identifier, the study date was emptied, and the name became the ANON^TRIAL the script asked for.

Three things the trial taught

Anonymizing makes a copy, and the original stays

After the run, Orthanc held six studies where three went in: the three originals with every identifying attribute intact, and the three anonymized copies. That's the documented behaviour for studies and series, and it's sensible for a hospital archive, which must keep the identified original. It also means "we anonymized it" says nothing about what's still on the disk.

For MARA that gives one rule: identified and de-identified images never share a store. The identified study belongs in the patient's own partition, under the patient-data audit. Only the de-identified copy may reach the cohort store that learns across patients, and the step that makes it is followed by a check that the original went where it should.

Labels are cleaned, and pixels stay as they were

All 24 anonymized slices had pixel data byte-identical to their originals; the trial compared hashes. The bright rectangle standing in for a burned-in name was still there. The file still said BurnedInAnnotation = YES, beside Orthanc's stamp of PatientIdentityRemoved = YES. The file now claims more than was done.

The DICOM standard does put burned-in text inside its basic profile. It leaves the hard part to whoever runs it. The 2026d edition says the de-identifier "should ensure" that no identifying information is burned into the pixels, either because the scanner never adds it or by removing it with the Clean Pixel Data Option. It adds that "the means by which burned in or graphic identifying information is located and removed is outside the scope of this Standard", and it warns that de-identifying the attributes "does not imply de-identification of the Information Object". Orthanc's

anonymize cleans attributes only, as its documentation says. The pixel duty stays with whoever deploys it. Ultrasound, screenshots and scanned documents often carry names and dates in the image itself.

MARA's design therefore adds a check when an image arrives. If BurnedInAnnotation is YES, or missing on a kind of image that often carries text, the image goes to quarantine, and a person clears it or a pixel-cleaning step is applied and verified. An image whose pixels haven't been cleared never carries a de-identified label into the cohort store.

Defaults are for a trusted room

Each refusal in the table came from a setting the trial turned on: login, the AE allowlist, the called-AE check and the loopback ports. None of them is on by default. Orthanc's guide is direct about whose job that is: "Orthanc is just software, security is your responsibility." pydicom showed the same lesson from the other side. When the synthetic script first built a frame-of-reference UID 66 characters long, two over DICOM's limit of 64, pydicom warned about it. MARA's intake should treat that kind of warning as a refusal rather than read the file anyway, because files from outside are where malformed and hostile input arrives.

From the trial to MARA's design

Figure 2 is the imaging path the trial suggests for MARA. It's a design. None of it is built yet.

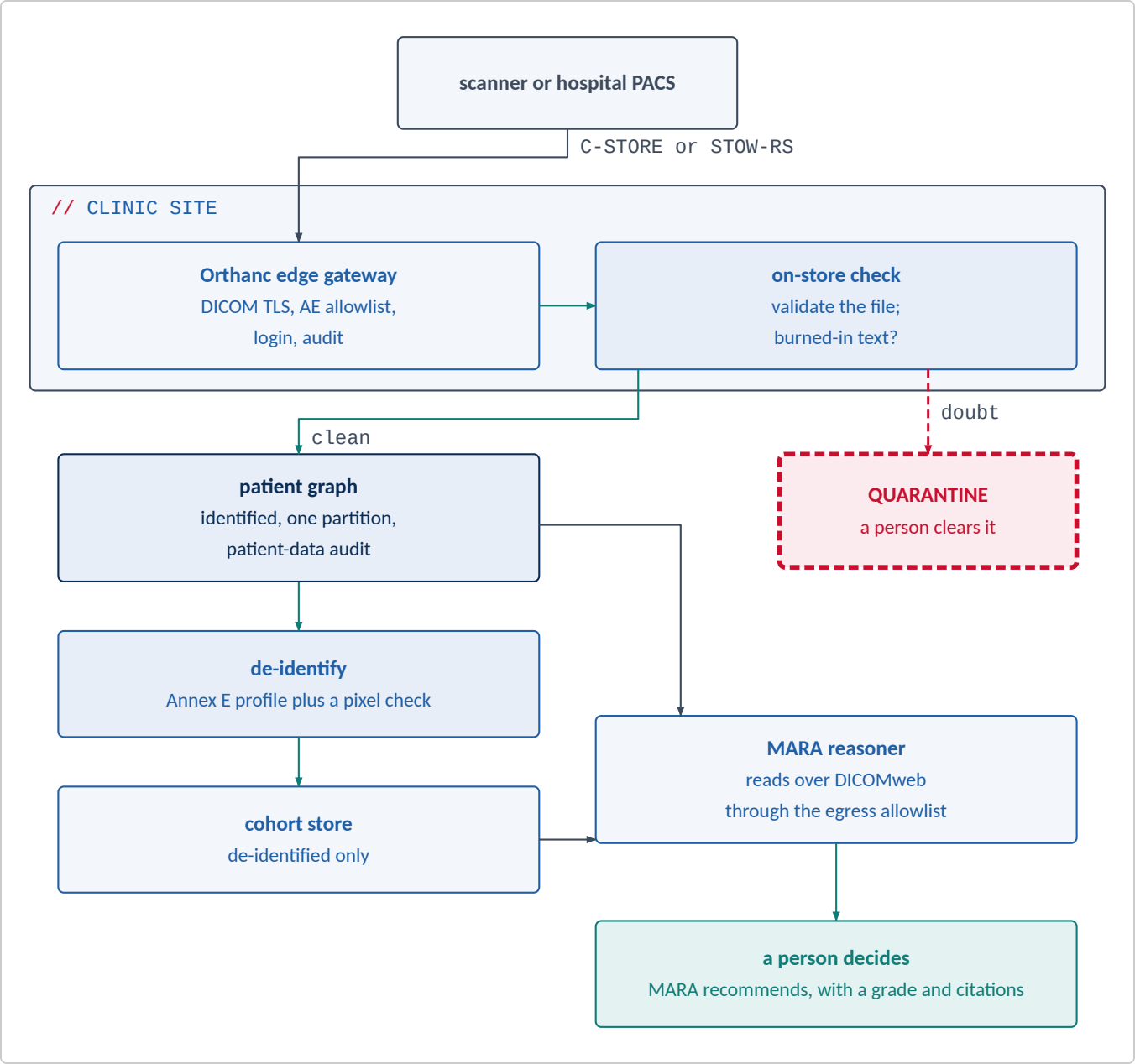


Figure 2. Follow an image down the page: doubt sends it to quarantine, identified images stay in the patient's partition, and only de-identified copies reach the cohort store before a person decides.

At each clinic, Orthanc would be the gateway. It receives images from scanners or the hospital's PACS over DICOM TLS or DICOMweb, accepts only listed senders, requires login and writes every receipt to the audit. An on-store hook, written with Orthanc's Python plugin, would validate each file and quarantine anything in doubt, and the clean images would join the patient's own record in MARA's patient graph. A separate step de-identifies them for the cohort store, checking pixels as well as labels. MARA's reasoner reads images over DICOMweb through the same egress

allowlist as every other source, and what she produces is a recommendation with its grade and citations, never an order. She reads from the clinical archive and never writes into it.

Two outside facts bound this design. De-identification under US law is a legal standard, and a file setting can't meet it alone. HIPAA, the federal health privacy law, allows two methods (45 CFR 164.514(b)). "Safe Harbor" removes 18 kinds of identifier, which include all elements of dates except the year, so a study date counts. The other is a qualified expert's determination that the risk of re-identification is very small. Following the DICOM profile is one part of meeting either; it isn't proof of either.

Software that manages or processes medical images for clinical use is also regulated. 21 CFR 892.2050 defines a "medical image management and processing system" as a device for "the review and digital processing of medical images for the purposes of interpretation by a trained practitioner". It's Class II. DICOM is named among its voluntary standards, and the FDA has cleared many PACS and image viewers under it. Orthanc in a trial on synthetic data is none of that. Neither is MAX4. A MARA that shows clinicians images to support a decision would need a regulatory assessment first.

What MAX4 trials next

Each of these can be done with synthetic data on MAX4, and each maps to a proof MARA needs:

- Encryption in transit: DICOM TLS between the test sender and Orthanc, which Orthanc has supported since 1.9.0.
- Quarantine: an Orthanc Python-plugin hook refuses or quarantines files with BurnedInAnnotation = YES, and the refusal is logged and counted.
- Who allowed it: every store, search and anonymization writes an audit entry with allowed_by and authority to MAX4's hash-chained log.
- The original, accounted for: after de-identification, a check confirms the identified original is where the rules say and nowhere else.

- Find and move: C-FIND and C-MOVE run between two Orthanc instances, standing in for a clinic gateway and a hospital archive.
- Realistic sizes: retrieval is timed on studies of hundreds of slices, using public, already de-identified research collections only, under each collection's own terms of use.
- Failures published: refusals, malformed files and quarantined images are reported beside the successes.

What isn't known

- How often real clinical images carry burned-in text their files don't declare is unknown. The trial's synthetic image declared it; a real one may not.
- How well an automatic pixel-cleaning step works, and how to verify it without a person looking at every image, is still open.
- What the AGPL licence on Orthanc's plugins requires of a MARA deployed to clinics needs counsel.
- Whether Orthanc is a sound gateway at clinical scale for MARA's use is untested. It's widely used, and this trial ran 24 small images.
- Which parts of MARA's imaging path would be regulated as a medical device, and in which countries, hasn't been assessed.

Why it matters

An archive that says "anonymized" can still hold every original, and an image that says "de-identified" can still carry a name in its pixels. Both are defaults, written down in the documentation, and either would have reached MARA's cohort store unnoticed if the first real scan had been her first scan. The trial found them on 24 invented slices, on a machine that holds no patient data, where finding them harmed no one. On a real patient's study, either default would have exposed an identity the file said was gone, and MARA would have learned from it.

Sources

- Orthanc Book, "Anonymization and modification" (REST endpoints, the default PS3.15 Table E.1-1 profile, DicomVersion 2023b default, KeepPrivateTags), orthanc.uclouvain.be/book/users/anonymization.html (read 25 October 2026).
- Orthanc Book, "Securing Orthanc", orthanc.uclouvain.be/book/faq/security.html (read 25 October 2026).
- Orthanc Book, "Licensing", orthanc.uclouvain.be/book/faq/licensing.html; Orthanc DICOMweb plugin README, orthanc.uclouvain.be/hg/orthanc-dicomweb/file/tip/README; Orthanc Book, "Python plugin", orthanc.uclouvain.be/book/plugins/python.html.
- Orthanc Book, "DICOMweb plugin", orthanc.uclouvain.be/book/plugins/dicomweb.html.
- Orthanc 1.13.0 publiccode.yml (release date 2026-08-15); the version was also reported by the running server's /system endpoint during the trial.
- Zenodo record 20798179 (June 2026), a description of the Orthanc ecosystem and its uses.
- DICOM Standard PS3.15 2026d, Annex E, "Attribute Confidentiality Profiles", dicom.nema.org/medical/dicom/current/output/chtml/part15/chapter_E.html (read 25 October 2026).
- pydicom on PyPI (3.0.2, 19 March 2026, MIT) and pynetdicom on PyPI (3.0.4, MIT); the pydicom tutorial "Dataset basics" and the v3.0.0 release notes (encoding follows the Transfer Syntax UID), pydicom.github.io.
- NVD, CVE-2026-32711 (pydicom DICOMDIR path traversal, fixed in 3.0.2), nvd.nist.gov/vuln/detail/CVE-2026-32711.
- 45 CFR 164.514(b), the HIPAA de-identification standard (Safe Harbor and Expert Determination), law.cornell.edu/cfr/text/45/164.514.
- 21 CFR 892.2050, law.cornell.edu/cfr/text/21/892.2050; FDA 510(k) clearance letters citing it, for example K232100, K223048 and K253639 (accessdata.fda.gov).
- MAX3 repository: MARA technical design (mara/spec/MARA_TECHNICAL_DESIGN.md), requirements from MAX

(mara/REQUIREMENTS_FROM_MAX.md), interoperability notes (mara/compliance/INTEROP.md), and MAX4's MARA proof and clinical foundation (desktopapp/docs/architecture/MARA_PROOF.md, CLINICAL_FOUNDATION.md).

- The trial itself: make_synthetic.py, trial.py, run_orthanc.sh, test_trial_scripts.py and trial-report.json, run 25 October 2026 on the image orthancteam/orthanc@sha256:d2705f2c….

Glossary

TERM	DEFINITION
AE title (application entity title)	the name a DICOM endpoint uses on the network. Servers can accept only names they know, which is how the trial refused STRANGER.
Burned-in annotation	text such as a name or date drawn into the image pixels rather than stored as a label. Label-based anonymization leaves it in place.
C-ECHO, C-STORE, C-FIND, C-MOVE	the DIMSE commands that test a connection, send an image, search and move studies.
De-identification	removing or replacing information that identifies a person. Under HIPAA it's a legal standard with two methods.
DICOM (Digital Imaging and Communications in Medicine)	the standard file format and network protocols for medical images.
DICOMweb (QIDO-RS, WADO-RS, STOW-RS)	DICOM services over HTTP for search, retrieval and storage. MARA's reasoner would read images this way.
DIMSE	the older DICOM protocol family, run between named application entities.
MARA (Medical Agentic Resident Assistant)	the Collective's design for a clinical assistant; she recommends, and a person decides.
MAX4	the desktop rebuild of MAX, from October 2026 on. It stores no patient data and is where MARA's substrate is proved first.
Orthanc	an open-source DICOM server, run in the trial as a small PACS on the loopback address.

TERM	DEFINITION
PACS (picture archiving and communication system)	the archive that stores medical images and serves them to viewers and systems.
Private tag	a vendor-defined attribute in a DICOM file. It can hold anything, including identifiers, which is why the anonymizer drops it.
pydicom	a Python library for reading, changing and writing DICOM files. The trial used it to build the synthetic studies and to check every anonymized copy.
UID	a globally unique identifier for a study, series or image instance. Left unreplaced, it can link a copy back to its source.

Appendix D, running the trial

The commands below rebuild the trial on any machine with Docker and Python. The password in run_orthanc.sh is a trial value for a loopback-only test server; change it for anything else.

pip install "pydicom==3.0.2" "pynetdicom==3.0.4" numpy pytest

python -m pytest -c /dev/null scripts # 5 offline tests of the scripts

./scripts/run_orthanc.sh # Orthanc on 127.0.0.1, login required

python scripts/make_synthetic.py ./synthetic # 3 invented patients, 24 slices

python scripts/trial.py ./synthetic # prints the JSON report

docker rm -f orthanc-trial # remove the test server and its data

HEALTHCARE ART-HLC-021

// ABOUT THE AUTHOR

Jason Newell leads organizations and people in technology, and is a researcher, deep technologist and builder — from a CRAY to agentic AI. He founded MAX Research Collective to build AI for regulated industries that runs where its owner has control, with healthcare as its strongest focus. About Jason → Connect on LinkedIn →

Newell, J. (2026). MAX Research Collective, ART-HLC-021. CC BY 4.0