Session S7: LNADA System Administration

Keeping the Catalog Running

Agenda for Today

- NADA platform architecture (10 min)
- Installation overview (20 min)
- Connecting Metadata Editor to LNADA (10 min)
- User management, roles & permissions (15 min)
- HTTPS & security best practices (15 min)
- Q&A (5 min)

Part 1: NADA Platform Architecture

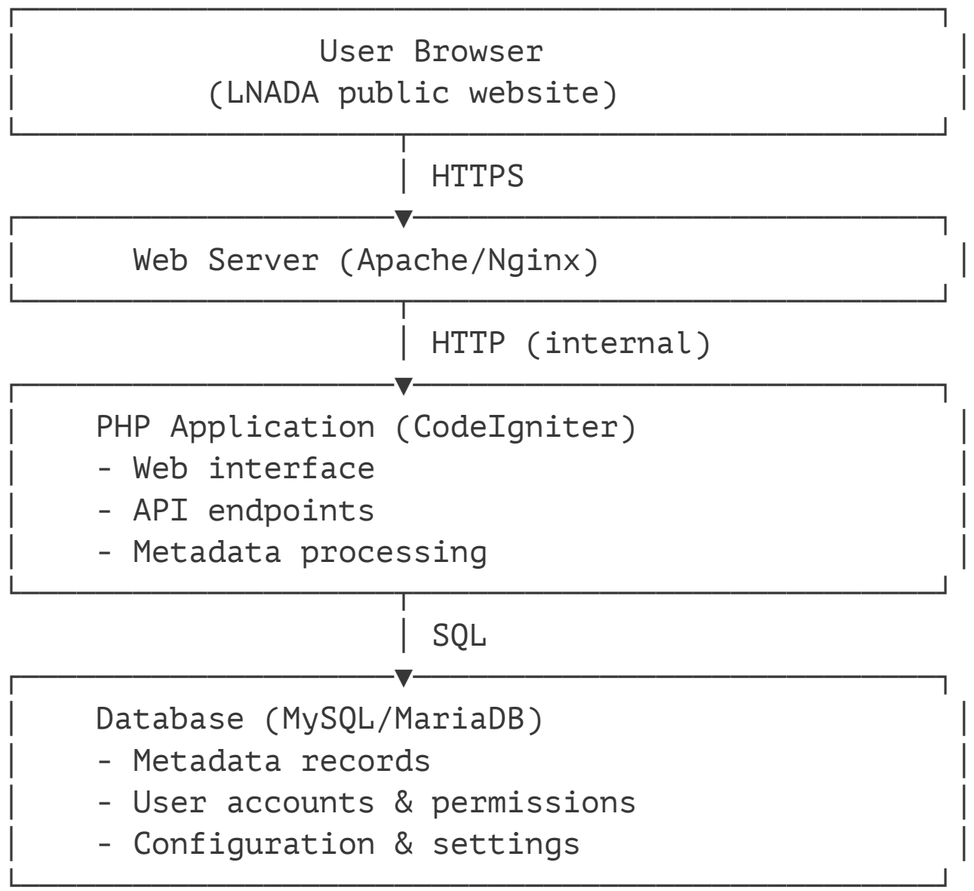
What is NADA?

NADA = National Data Archive
- A data catalog and dissemination platform
- Open-source software (developed by IHSN)
- LNADA = Lao PDR's NADA instance
- Used globally by national statistics offices and research institutions

NADA 5.x Technology Stack

Component	Technology	Version
Web Framework	PHP + CodeIgniter	PHP 8.1+
Database	MySQL or MariaDB	8.x
Web Server	Apache or Nginx	2.4+
Operating System	Linux	—

System Architecture Diagram



Metadata Editor Integration

Two separate applications:

1. **NADA** (this session) — public catalog & dissemination

2. **Metadata Editor** — metadata creation tool

How they communicate:
- Metadata Editor → NADA via **REST API**
- Metadata editors create/edit metadata in the Editor
- Editor publishes to NADA automatically
- NADA stores metadata and makes it searchable

Part 2: Installation Overview

Important: This is NOT a Live Exercise

We are **walking through** the installation process for reference.
- Full installation takes 2–4 hours
- Requires server administration experience
- LNADA is already installed on LSB/MOF servers
- This knowledge is useful for: troubleshooting, updates, migration

Server Prerequisites (1/2): Hardware

Hardware Requirements:

- CPU: Dual-core or better
- RAM: 4GB minimum, 8GB recommended
- Disk: 20GB minimum (more for large catalogs)
- Network: Stable internet, static IP recommended

Server Prerequisites (2/2): Software

Software Requirements:

- **Operating System:** Linux (Debian, Ubuntu, CentOS)
- **Web Server:** Apache 2.4+
- **PHP:** Version 8.1 or newer
- **PHP Extensions:** xsl, xml, mbstring, mysqli
- **Database:** MySQL 8.x or MariaDB

NADA Installation Steps (1/2)

1. Download from GitHub

```
bash
git clone https://github.com/ihsn/nada.git /var/www/html/nada
```

2. Create Database

```
bash
mysql -u root -p
CREATE DATABASE nada_db;
CREATE USER 'nada_user'@'localhost' IDENTIFIED BY
'strong_password';
GRANT ALL PRIVILEGES ON nada_db.* TO 'nada_user'@'localhost';
```

NADA Installation Steps (2/2)

3. Set Folder Permissions

```
bash
chmod -R 775 /var/www/html/nada/application/cache
chmod -R 775 /var/www/html/nada/datafiles
chmod -R 775 /var/www/html/nada/files
```

NADA Installation Steps (3/3)

4. Run Web Installer

- Navigate to: `http://your-server/nada/web_installer.php`
- Enter database credentials and admin email/password

5. Create First Admin Account

- Login to NADA admin panel → Users → Create admin
- Assign to yourself or designated administrator

6. Verify Installation

- Test: `http://your-server/nada/`
- Test admin panel: `http://your-server/nada/admin/`

Metadata Editor Installation

Two separate components to install:

1. Metadata Editor PHP Application
- Repository: github.com/ihsn/editor
- Location: `/var/www/html/editor/`
- Database: `metadata_editor` MySQL database
- Config file: `database.php`

2. PyDataTools (Python/FastAPI)
- Repository: github.com/mah0001/pydatatools
- Purpose: Automatic metadata extraction from data files
- Requirements: Python 3.12+, pip, uvicorn

Metadata Editor Setup (1/2)

Install PyDataTools:

```
bash
git clone https://github.com/mah0001/
pydatatools.git
cd pydatatools
pip install -r requirements.txt
uvicorn main:app --host 0.0.0.0 --port 8000 &
```

Metadata Editor Setup (2/2)

Configure permissions:

```
bash
chmod -R 775 /var/www/html/editor/datafiles
chmod -R 775 /var/www/html/editor/files
chmod -R 775 /var/www/html/editor/logs
```

Run web installer at: `http://your-server/editor/index.php/install`

Recommended: Container-Based Deployment with Dokploy

The manual installation shown above is for reference only.

For production, we recommend **Dokploy** (dokploy.com):
- Open-source, self-hosted server management platform
- Container-based deployment (Docker) — reproducible & portable
- Web UI for managing applications, databases, SSL certificates
- Easy backups, restarts, and scaling
- One-click deployments and rollbacks

Why containers over manual installation?
- Manual setups are hard to replicate and debug
- A failed server means starting from scratch
- Containers can be restarted, migrated, or rebuilt in minutes

Part 3: Connecting Metadata Editor to LNADA

API Communication

The Metadata Editor needs to "know about" NADA:

1. Where NADA is located (URL)
2. How to authenticate (API key)
3. Permission to publish metadata

Configuration Steps (1/2)

In NADA Admin Panel:

1. Users → API Keys
2. Create service account (e.g., `metadata-editor`)
3. Generate and copy API key
4. Paste into Editor settings

Configuration Steps (2/2)

In Metadata Editor:

1. Site Administration → Settings
2. Enter NADA URL: `https://nada.lsb.lao-stat.de/`
3. Paste the API Key from NADA
4. Test connection → Success message

Verify the connection:

- Publish a test metadata entry from the Editor
- Verify it appears in LNADA public catalog

Troubleshooting API Connection

Common issues:

Problem	Solution
"Connection refused"	Check NADA URL is reachable; firewall rules
"Authentication failed"	Verify API key is correct and active
"Timeout"	Check network bandwidth; NADA server load
"Bad request"	Verify URL format (with trailing slash if needed)

Part 4: User Management, Roles & Permissions

Users in NADA

NADA has 4 user roles:

Role	Permissions	Use Case
Site Admin	Full system access, user management, config	LSB IT, lead data manager
Collection Admin	Manage specific collection, users in collection	Department heads managing datasets
Reviewer	Review & approve metadata submissions	Quality assurance staff
General User	View published metadata, download (if allowed)	Researchers, public, LSB staff

Users in Metadata Editor

Metadata Editor has 3 user roles:

Role	Permissions	Use Case
Admin	Create users, manage projects globally	Metadata manager
Editor	Create & edit projects, publish to NADA	Metadata documenters
Viewer	View-only access, no editing	Observers, auditors

Best Practices: Roles & Permissions (1/2)

1. **Principle of Least Privilege**
    - Give users minimum permissions needed
    - Don't make everyone admin
2. **Separate Admin Accounts**
    - One personal account for daily work
    - One admin account for admin tasks
    - Use admin sparingly

Best Practices: Roles & Permissions (2/2)

1. **Regular Password Rotation**
   - Change admin passwords every 90 days
   - Require strong passwords (12+ characters)
   - No password reuse (last 5 passwords)
2. **Audit Logs**
   - Monitor who did what and when
   - Review suspicious activity
   - Keep logs for 1+ years

Part 5: HTTPS & Security

Why HTTPS is Mandatory

Without HTTPS (HTTP):
- User credentials sent in plain text
- Metadata visible to network eavesdroppers
- Public doesn't trust your catalog

With HTTPS (encrypted):
- Credentials protected in transit
- Metadata access logs protected
- Professional, trustworthy appearance
- Required by modern browsers

Setting Up HTTPS with Let's Encrypt (1/2)

Install certbot:

```
bash
apt-get install certbot python3-certbot-apache
certbot certonly --apache -d nada.lsb.lao-stat.de
```

Configure Apache VirtualHost:

```
apache
<VirtualHost *:443>
    ServerName nada.lsb.lao-stat.de
    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/...
    SSLCertificateKeyFile /etc/letsencrypt/live/...
</VirtualHost>
```

Setting Up HTTPS with Let's Encrypt (2/2)

Auto-renewal (add to cron):

```
bash
certbot renew --quiet
```

Let's Encrypt certificates expire every 90 days. Schedule this to run daily.

PHP Security Hardening (1/2)

Edit `php.ini`:

```ini
; Hide PHP version from headers
expose_php = Off

; Never display errors to users
display_errors = Off
log_errors = On
error_log = /var/log/php-error.log

; Disable dangerous functions
disable_functions = exec,shell_exec,system
```

PHP Security Hardening (2/2)

File upload and session settings:

```ini
; Limit file uploads
upload_max_filesize = 50M
post_max_size = 50M

; Session security
session.cookie_httponly = On
session.cookie_secure = On
```

Database Security (1/2)

Create dedicated, unprivileged user:

```bash
```

CORRECT: Least privilege

```
CREATE USER 'nadauser'@'localhost'
 IDENTIFIED BY 'strongpassword';
GRANT SELECT, INSERT, UPDATE, DELETE
 ON nadadb.* TO 'nadauser'@'localhost';
```

Database Security (2/2)

MySQL security hardening:

```
bash
mysql_secure_installation
```

This removes:

- Anonymous users
- Remote root login
- Test databases

Additional Security Measures (1/2)

Firewall & Network:

- Only open necessary ports (80, 443)
- Restrict admin panel access to LSB IPs
- Use VPN for off-site admin access

File Permissions:

- Data files: `755` (world-readable)
- Config files: `600` (admin-only)
- NADA application: `755`

Additional Security Measures (2/2)

Regular Patching:

- Subscribe to GitHub security releases
- Test updates in staging first
- Apply to production within 48 hours
- Keep OS patched: `apt-get update && upgrade`

Backup Encryption:

- Encrypt backups before storing off-site
- Store encryption keys separately

Part 6: Q&A

Questions on System Administration?

- Architecture & installation
- Metadata Editor integration
- User management & roles
- Security & HTTPS setup
- Backup & disaster recovery

Resources:
- NADA Installation Guide: ihsn.github.io/nada-documentation/installation-guide
- NADA Admin Guide: ihsn.github.io/nada-documentation/admin-guide
- Metadata Editor Docs: worldbank.github.io/metadata-editor-docs

Wrap-Up: Session S7

You now understand:
- How NADA is built (PHP, MySQL, Apache)
- Installation requirements & process
- How Metadata Editor connects to NADA
- User roles and best practices
- HTTPS and security hardening
- Importance of backups (covered in S8)

Next session (S8):
- Site configuration walkthrough
- Backup & disaster recovery (crucial!)
- System maintenance & updates
- Complete wrap-up