



**MALMÖ HÖGSKOLA**

# **Paket- och beroendehantering med Composer**

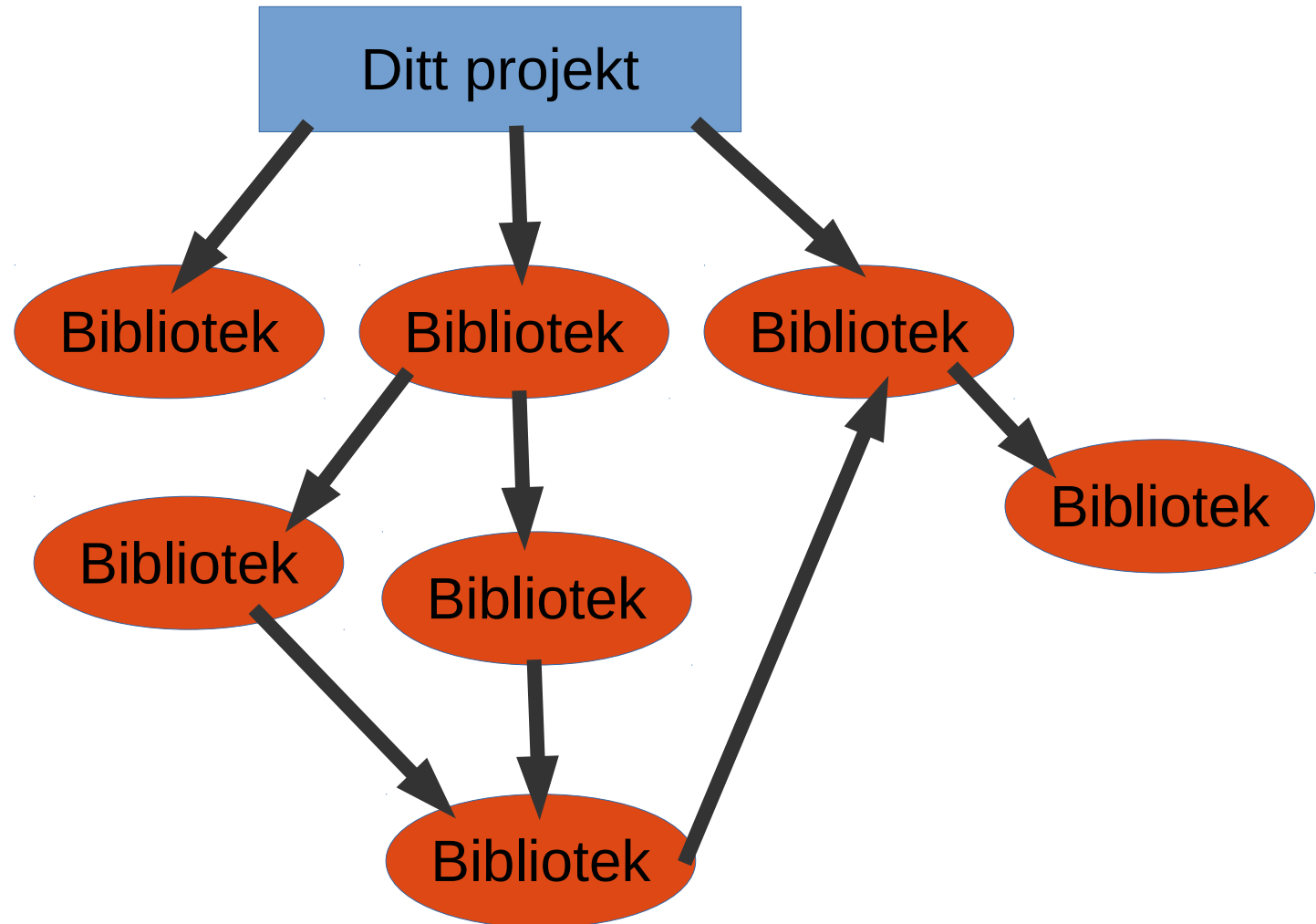
# Dagens agenda

- Vad är paket- och beroendehantering?
  - Pakethantering i operativsystem
  - Beroendehantering i mjukvaruprojekt
- Beroendehantering i PHP
  - Introduktion till Composer
  - Packagist
  - Semantisk versionering
  - Autoloading
- Exempel

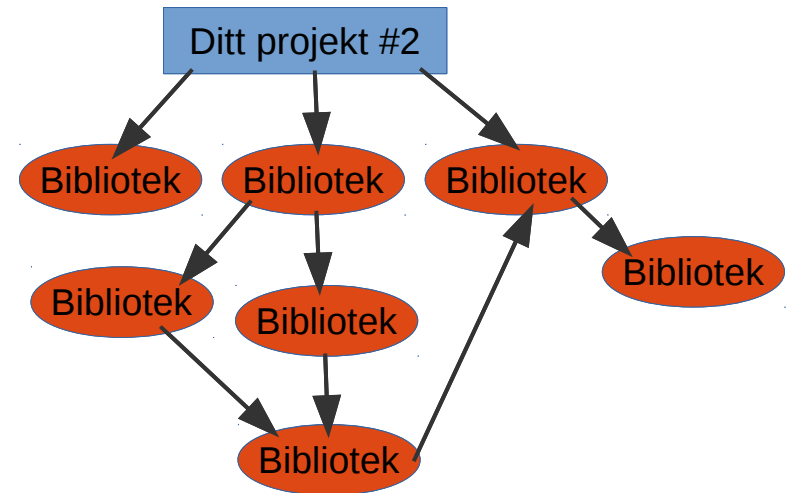
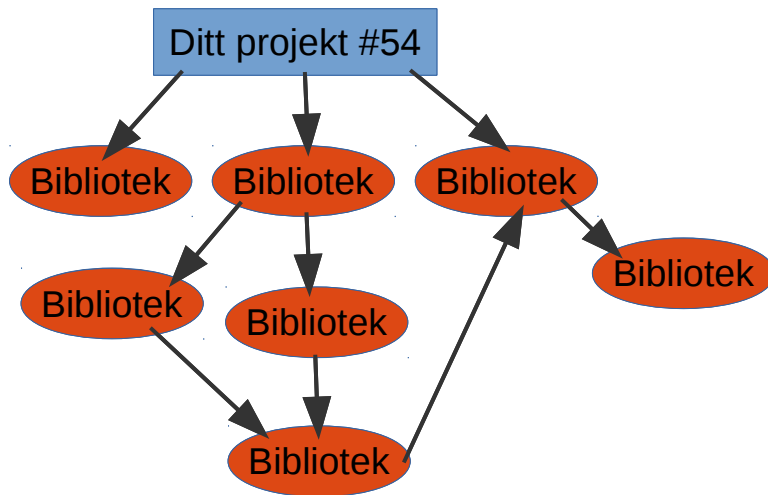
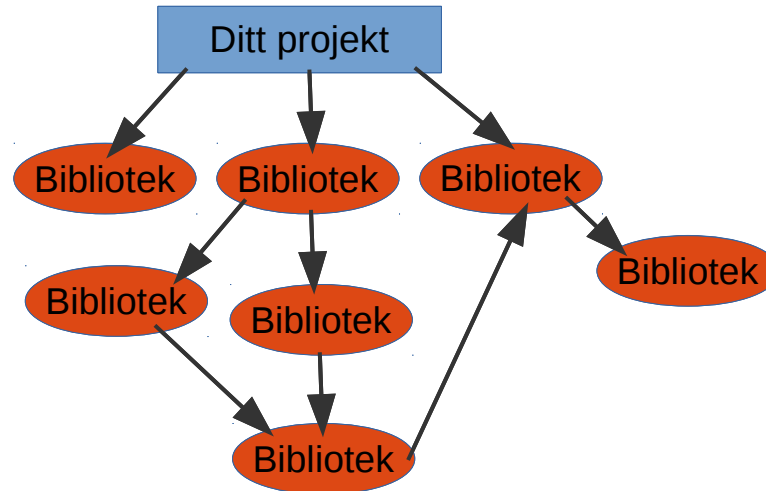
# Problem



# Beroenden



# Flera projekt



# Pakethantering i operativsystem

Exempel: apt-get i Ubuntu

# Beroendehantering i mjukvaruprojekt

- Python: pip
- Java: Maven/Gradle
- C#/.NET: NuGet
- **PHP: Composer**

# Varför beroendehantera i mjukvaruprojekt?

- Robustare projekt och mjukvara
- Enklare att starta nya projekt
- Enklare för nya utvecklare att börja arbeta
- Mindre kod i era repositories



# Composer



Composer: <https://getcomposer.org>  
Lockart, *Modern PHP*. p. 57

# Composer

- De facto-standard för vettig beroendehantering i PHP
- Hanterar beroenden i flera led
- Kan skilja mellan olika versioner av paket
- Sköter installation av paketen åt dig


Composer: <https://getcomposer.org>  
Lockart, *Modern PHP*. p. 57

# Composer – installation

Exempel: installation av Composer i Ubuntu

# Packagist

**Packagist** The PHP Package Repository[Browse](#)[Submit](#)[Create account](#)[Sign in](#)



Packagist is the main [Composer](#) repository. It aggregates public PHP packages installable with Composer.

## Getting Started

### Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package2": "1.*",
    "vendor/package3": "^2.0.3"
  }
}
```

For more information about packages versions usage, see the [composer documentation](#).

### Install Composer In Your Project

Run this in your command line:

```
curl -sS https://getcomposer.org/installer | php
```

Or [download composer.phar](#) into your project root.

See the Composer documentation for complete [Installation Instructions](#) on various platforms.

### Install Dependencies

Execute this in your project root.

```
php composer.phar install
```

### Autoload Dependencies

## Publishing Packages

### Define Your Package

Put a file named `composer.json` at the root of your package, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "description": "A short description of what your package does",
  "require": {
    "php": "^5.3.3 || ^7.0",
    "another-vendor/package": "1.*"
  }
}
```

This is the strictly minimal information you have to give.

For more details about package naming and the fields you can use to document your package better, see the [about](#) page.

### Commit The File

You surely don't need help with that.

### Publish It

[Login](#) or [register](#) on this site, then hit the [submit](#) button in the menu.

Once you entered your public repository URL in there, your package will be automatically crawled periodically. You just have to make sure you keep the `composer.json` file up to date.

**Packagist:** <https://packagist.org>  
Lockart, *Modern PHP*. p. 55

# Packagist

- Composers repository för PHP-paket
- Paketen är fria att använda i dina projekt
- Använder semantisk versionering för att skilja mellan olika versioner av paket

# Semantisk versionering

1 . 0 . 0 . b1

major minor micro qualifier

Semantisk versionering: <http://semver.org/>  
Lockart, *Modern PHP*. p. 61

# Autoloading

```
<?php
require __DIR__ . '/vendor/autoload.php';

$log = new Monolog\Logger('name');
$log->pushHandler(new Monolog\Handler\StreamHandler('app.log', Monolog\Logger::WARNING));
$log->addWarning('Foo');
```

Autoloading: <http://www.php-fig.org/psr/psr-4/>  
Lockart, *Modern PHP*. p. 63

# Composer – ett exempel

Exempel: Skapa ett Composer-projekt