

Creating a Custom Registration Process for Spring Security

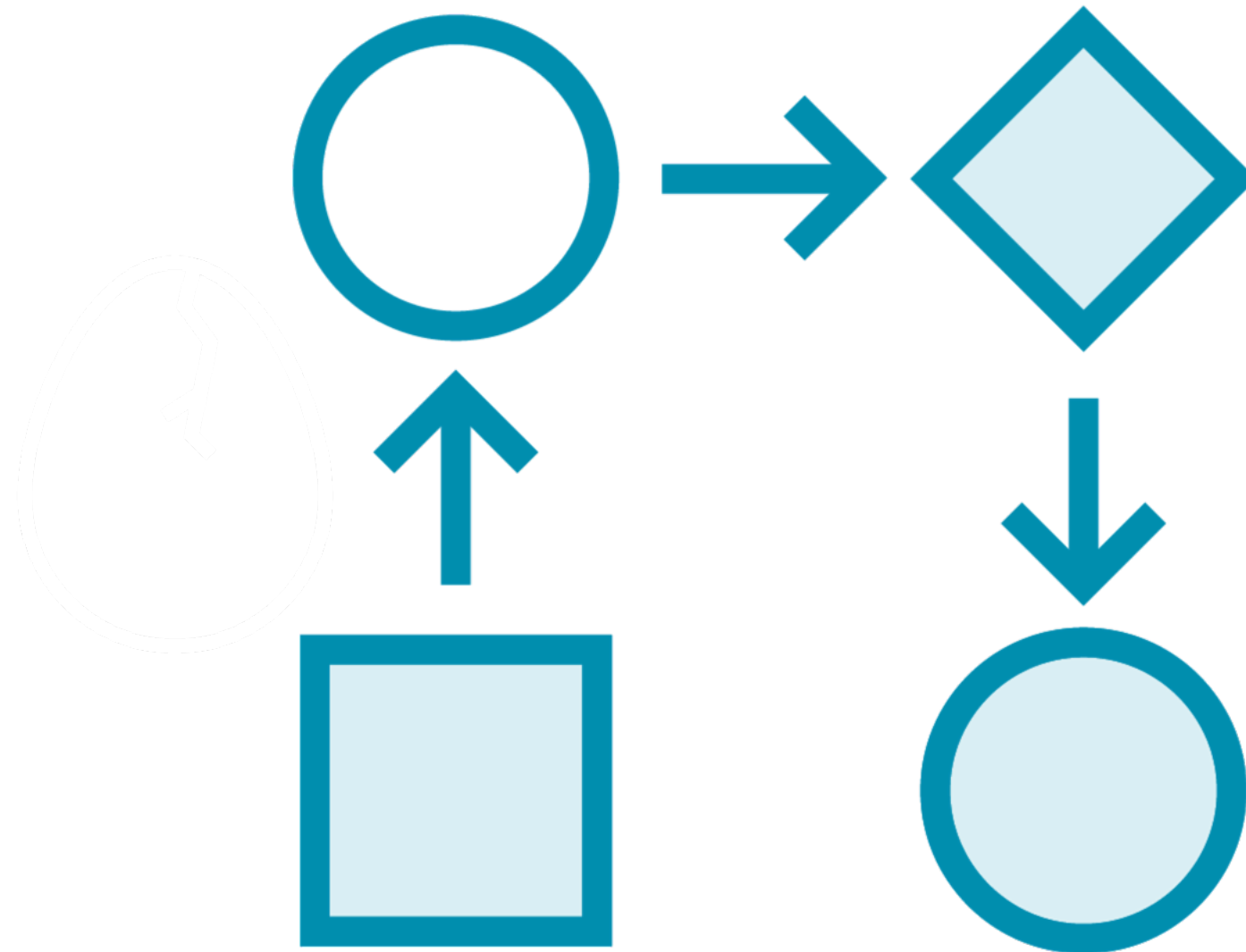


Bryan Hansen

DIRECTOR OF SOFTWARE DEVELOPMENT

@bh5k

Registration Workflow



Authentication



Email



Account UI

```
<form:form modelAttribute="account">
  <form:errors path="*" cssClass="errorblock" element="div" />
  <div><label> Username : <input type="text" name="username" /> </label></div>
  <div><label> First name : <input type="text" name="firstName" /> </label></div>
  <div><label> Last name: <input type="text" name="lastName" /> </label></div>
  <div><label> Email: <input type="text" name="email" /> </label></div>
  <div><label> Password: <input type="password" name="password" /> </label></div>
  <div><label> Confirm Password: <input type="password" name="matchingPassword" /
> </label></div>
  <input type="submit" class="btn btn-lg btn-primary" role="button"
value="Submit" />
</form:form>
```

Account Controller

```
@GetMapping("account")  
public String getRegistration(@ModelAttribute  
("account") Account account) {  
    return "account";  
}
```

```
@PostMapping("account")  
public String addRegistration(@Valid @ModelAttribute  
("account") Account account, BindingResult result) { //...
```

Account Service

```
package com.pluralsight.conference.service;

import com.pluralsight.conference.model.Account;

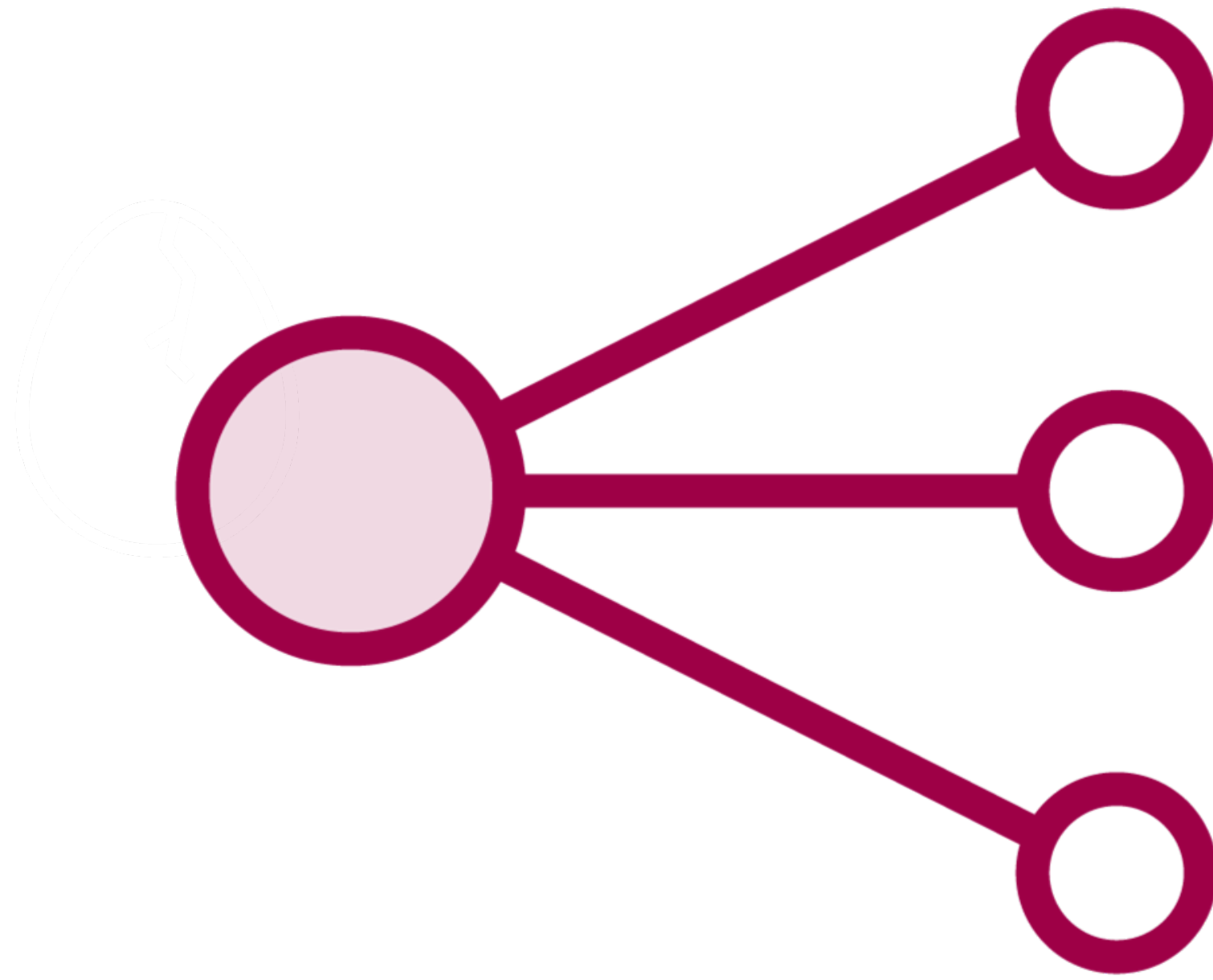
public interface AccountService {

    public Account create (Account account);

    void createVerificationToken(Account account, String token);

    void confirmAccount(String token);
}
```

Send Mail Event



Confirm Account

```
<div class="alert alert-success" role="alert">  
    Your account has been confirmed.  
</div>
```

Summary

Create Account

Asynchronous Email

Verification Token

Confirmation Process

Cleanup