



# Acumatica Developer

## XRP Priorities & Roadmap

Mikhail Shchelkonogov  
Founder & Chief Technology Officer  
[acumatica.com/developers](https://acumatica.com/developers)

# Key Initiatives for Acumatica XRP

---

- ✓ Continuous platform technology advances
- ✓ Improved UI and usability
- ✓ Non-programming customization
- ✓ Dashboard and reporting enhancements
- ✓ Machine learning and artificial intelligence
- ✓ Performance and scalability

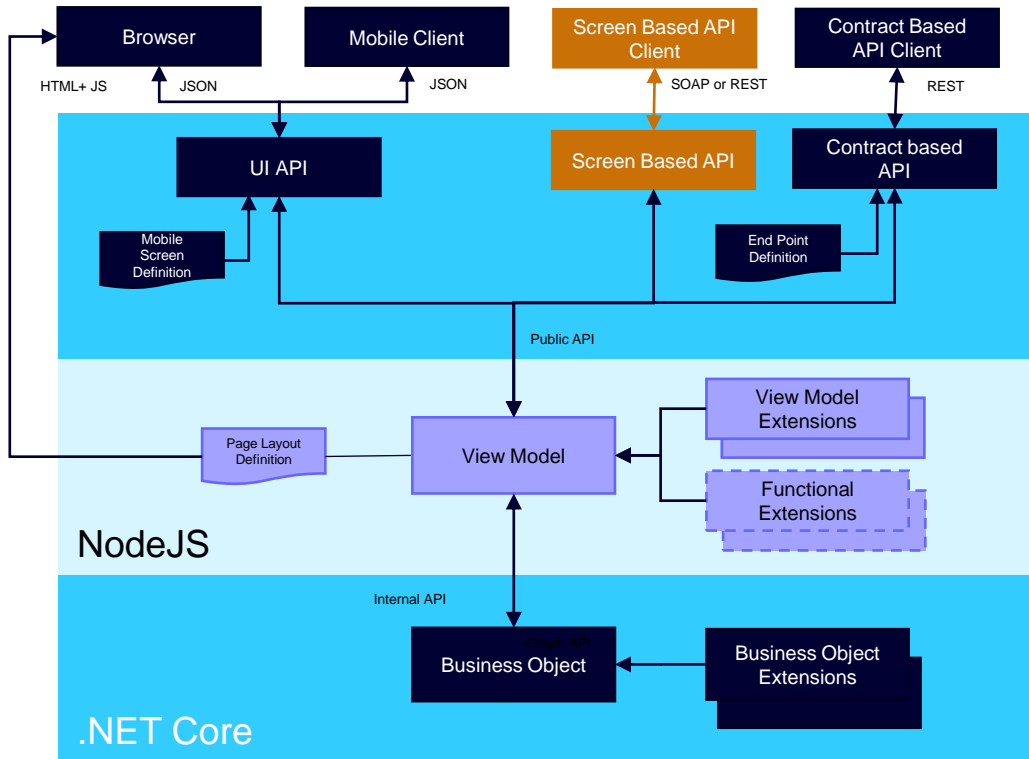


# Continuous platform technology advances

---

- Moving Acumatica to .NET core platform
- Moving common application logic to extensions
  - Multi currency support
  - Tax calculation
  - Discount and price calculation
  - Inventory allocation
- Integrating with external authentication providers and APIs
  - OpenID Connect support
  - Microsoft Graph API support
- New UI framework
- Programming API improvements

# New UI Framework



- Client side page rendering
- Aurelia.io as a frontend platform
- View Model declaration on TypeScript
- Page layout definition on HTML
- TypeScript functional extensions

# Improved UI and usability

---

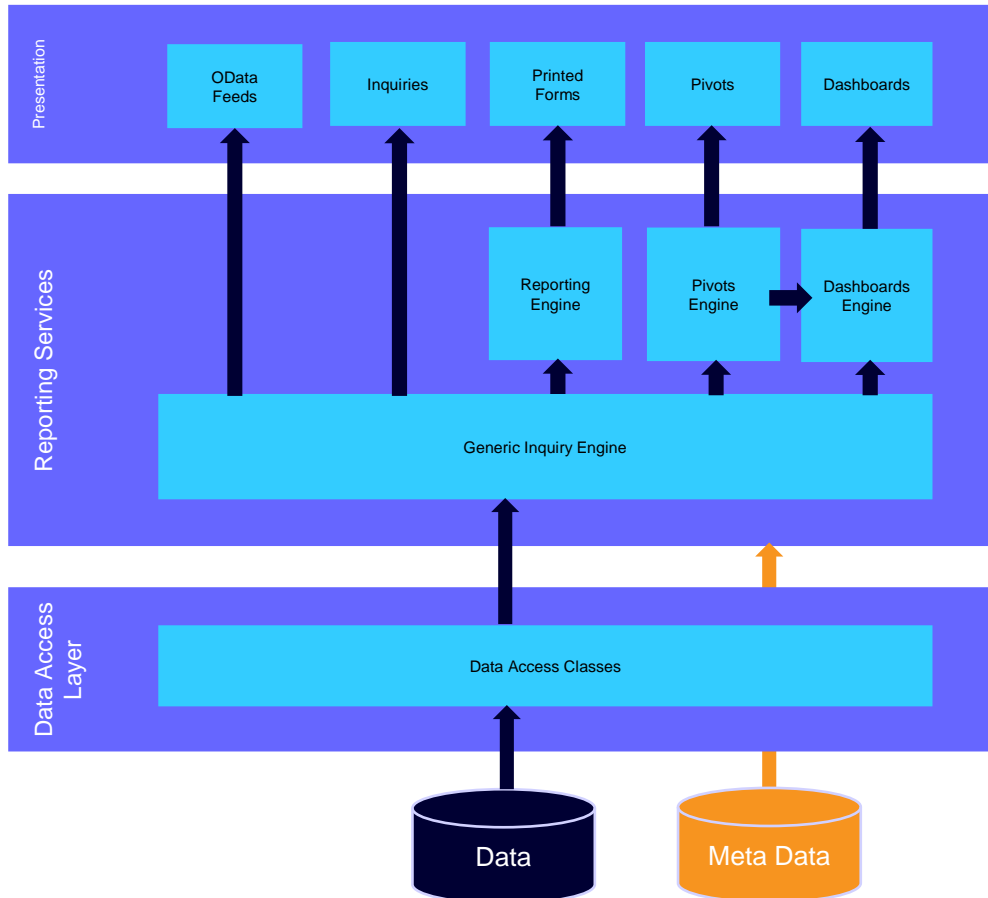
- New Toolbar
- New Grid
- New advanced filtering
- Non - blocking errors messages
- Quick record view
- In place record editing
- ... and much more to come

# Non-Programming Customization Framework

---

- More UDF Functionality
  - Defaults
  - Custom related lookups
  - Formulas
  - Inheritance of UDF fields between business objects
- Business Events and Automation
  - Create Task and Activity
  - Improved Reports Scheduling
- New visual form customization designer
- New type script customization layer

# Dashboard and reporting enhancements



- Move towards Common Reporting Architecture
- Decoupling data queries from the presentation in Generic Inquiry
- Moving report engine on top of Generic Inquiry
- New Generic Inquiry visualizations
  - Grid grouping with subtotals
  - Charts
  - Reports
- Improved dashboard parameters
- New dashboard visualization

# Performance & Scalability

---

- New SQL governing functionality that limit requests with excessive result sets
- Various query cache optimizations and elimination of excessive and unnecessarily SQL requests
- Enhanced Store Cached API
- Async support in the database layer
- Elimination of running totals from the historical tables
- Optimization of CPU and Memory consumption by Generic Inquiry and Dashboards
- Moving towards Docker & Kubernetes for better isolation and maintainability