Client-side and server-side experimentation both offer distinct advantages depending on the needs of your organization. Use our guide to determine which method—or combination of methods—of running experiments is right for your team.

**Client-Side Web**
- Client-side experimentation through a Javascript snippet.
- Marketing or growth teams who want to enable anyone on your team to create experiments with a visual editor, without the need for code releases.

**Server-Side Optimizely Full Stack**
- Server-side experimentation through developer SDKs.
- Product development teams who want to experiment deeply in a product, with minimal performance impact, and across multiple channels.

**Client-Side vs. Server-Side Experimentation?**

**ACCESSIBILITY**
- **Accessible to everyone.**
  - Client-side experimentation is the fastest way for teams to experiment, with a WYSIWYG visual editor, providing access to non-technical members of the organization.
- **Developers required.**
  - Server-side experimentation is the fastest way for server-side teams, since developers are needed to set up the experiments that run on nodes that are developed on company servers.

**DEPTH**
- **Focused on improving interfaces.**
  - Client-side experimentation is the fastest way to test how an experience looks and feels: layouts, messaging, colors, and more.
- **Deep experimentation in any application.**
  - Server-side experimentation can be used to test how a product works, including underlying features, business transactions, algorithms, etc.

**GOALS**
- **Accessible to everyone.**
  - Many experiments can be created using a WYSIWYG visual editor, providing access to non-technical members of the organization.
- **Developers required.**
  - Any new experiment code must be deployed through a server code release, subject to standard practices like code review, QA, monitoring, etc.

**CHANNELS**
- **Accessible to everyone.**
  - Client-side implementations only allow for experimentation in a single client, e.g. JavaScript for websites.
- **Developers required.**
  - Experiments run on the server can be executed on any channel, or span multiple channels at once, e.g. web, mobile, SMS, and email.

**IMPLEMENTATION**
- **Simple implementation.**
  - Client-side experimentation is the fastest way to experiment, with one line of JavaScript to empower experimentation site-wide.
- **Customized to technology stack.**
  - Integrating an experimentation solution may take more work, depending on the complexity of the server-side technology stack.

**CACHING**
- **No cache misses required.**
  - New experiments can be controlled and deployed without using a cold release through Optimizely airPaper.
- **Treatments must be cached at CDN layer.**
  - Server-side implementations will only work if treatments are determined at CDN layer. A CDN integration or separate decision service is needed.

**PERFORMANCE**
- **No performance impact.**
  - Experiment decisions and experiment execution can be done entirely in server side code, with no noticeable impact on performance to the end user.
- **Requires attention.**
  - Synthetics experiments can increase page or link loading times, especially for low-priority pages, which can cause a user to abandon the site.

Get started with client-side or server-side experimentation today at optimizely.com/getting-started.