Annex 1 –
Request for Proposals (RFP)
for Chess ID Game Viewer

Introduction:
FIDE is the organizer of the major chess events worldwide for top and amateur chess players with vast local and Internet audience. Internet audience makes up the largest part of the visitors for event and most of chess lovers prefer to follow chess events via website of the event.

Internet website provides richer media experience comparing to the local playing hall visit in terms of freedom of selection of certain games to follow, listening to preferred commentator, select chess analysis tools, view live games statistics immediately, store games for later review or view recorded games and commentary.

All major companies and online platforms on chess market use their own 'game viewer' applications to create maximally useful interface for viewers to follow live broadcasts and attract user on their websites.

During last years, FIDE used the 'game viewer' application from the above-mentioned companies and platforms on different collaborative levels and is highly depending on the 3rd party 'game viewer' applications for FIDE official events. This situation prevents FIDE from full collaboration with organizers and sponsors as FIDE cannot fully control the broadcasts of live chess games.

FIDE can provide consultation support during all development process.

The developer should provide timeline budget and working hours required.

Aims:
To provide FIDE with the independent technology to broadcast FIDE- and not FIDE-organized events in Internet with maximum efficiency and increase the media impact of chess tournaments by providing reach media experience to viewers of chess tournaments.

The proposal to develop a 'game viewer' application on the level of the existing solutions from the most prominent online chess platforms/broadcasters which will be designed according to needs of FIDE for the events and can be customized for each event and sponsors separately without dependence from other parties.

Concept and main idea:
Chess ID Game Viewer shall be a browser-run application enabling the display of live and recorded chess games in PGN format (txt based) to be displayed in user friendly and custom format to world chess audience taking into consideration existing solutions and aiming at top visual and technical performance.
It shall consist of client module and server module for analysis, data storage and communication with client.
It is highly recommended that developer checks existing solutions on market.
FIDE expects the product to be developed in the period of 4-5 month with timeline agreed with the developer and FIDE according to stages below.

Stages:
1. Creation of design of the board and clocks, taken pieces, pieces
2. Creation of design of the moves panel
3. Creation of design of the analysis area, advantage panel
4. Creation of design of full set in 1 module: chessboard, moves panel, analysis area and **video embed** in several variants – color schemes and positioning
5. 'Minimal' mode design creation with board and clocks only without analysis panel, short players data for stacked view on devices to fit maximum number of games per screen.
6. Approval of above-mentioned design variants.
7. Programming features and testing
8. Presentation of the product

Requirements:
Application should be optimized to run on clients in all major browsers (Chrome, Firefox, Safari) and major OS – Windows, MacOS, IOS, Android for long period of more than 8 hours without high memory usage and without slowing down the running device after running for several hours.

Preferable technology to be selected by developer to be optimized for sending small chunks of data by push or other method (1 game in 10-30Kb usually and only updates can be sent so size can be 5-10Kb), fast data storage/cache. FIDE estimates that system can have up to 10 M concurrent connections with idea that one connection is one game so one user can generate several connections.

The approach and exact technology used can be discussed with developer.

Setup will require knowledge of PGN, FEN, UCI - reference Wikipedia for exact format description.

General application work process stages:

Server:
Server receives data in PGN format from remote link via https. Other methods can be provided as addition.
- copies to local system
- checks the integrity of file
- analyses game with chess engine (split by FEN positions and then analyzes via UCI communication with chess engine) in several threads for faster performance.
- puts chess engine analysis into local fast storage - several analyses lines per move (position)
- checks clocks settings in file and updates local timers. Has its own timer in case clock times disappear from PGN file and outputs clocks data to client for display
- outputs PGN file and analysis lines, comments to client for display
- receives moves by commentator and inputs them as analysis lines into storage and outputs them for clients.

Client
- receive PGN file with moves, analysis and clock times generated by server
- displays corresponding data on boards and in panels
- panels: board with notation, moves panel with notation, taken pieces panel, analysis panel with lines generated by engine, clocks panel, advantage graphics panel.

General key requirements for system:
- read remote PGN file(s)
- minimal delay between the move in PGN and its display on the board not more that 10 seconds for live without analysis and 20 seconds with analysis.
- all panels, boards should scale proportionally on desktop and mobile
- internet connection problems should be reported
- high level error handling for incomplete data read – not readable data, incomplete data, not updated data, not complete game.
- no module should halt and 'freeze' in case of any errors not to require reloading of the page
- compare existing data for game and new data received and reload data completely if needed (server's module) to re-read last moves, all moves or result.
- read data even if game finished and check if any data was updated

Client module description:
Displays chessboards with clocks reading data from server and displays chess moves on the board, emulated chess clocks 'running' for each player. Show analysis lines with moves variations.
Should be able to include up to 24 boards on 1 screen of in full HD resolution. Mobile-friendly interface.
- adjustable size
- show clocks for white and black with running time
- highlight moves squares by settings or by clicks of a mouse (settings)
- provide animation of moves (settings)
- allow sound on moves update (settings)
- show current position on the board upon click on any move in a moves list of variations or moves panel
- notifies if moves shown is an actual move or move played from variant proposed by chess engine
- shows time spent for each move (settings)
- shows name, country, rating, photo of black and white players
- calculates clock values even if clock data stops in PGN file
- constant configuration reading and assigning states to ones given in configuration
- loads games dynamically via ajax
- several instances running on 1 page of a browser
- use javascript engine for analysis - not take analysis from server module (settings)
- switch board (rotate board) (button)
- show list (icons) of taken pieces (button)
- show advantage panel timeline (graphics vertical or horizontal) for game
- play moves automatically (button)
- replay whole game (button)
- go to the first move (button)
- show which color is to move
- show result
- show picture of the board position for each move

Commentator mode (second analysis board) description:
- move pieces by clicking
- input and save variations to server
- hotkeys to draw arrows on the board and highlight squares (e.g. pressing Ctrl+click)
- show analysis board separately
- play lines provided by chess engine on analysis board

- show list of available games from PGN file
- highlight the board with the activity made since the game was last viewed

Requested Options via configuration and ‘Settings’ menu for each instance:
- update frequency in ms
- visual size of modules
- selection of one of the several variants of graphics prepared
- each panel can be hidden (moves, analysis, photos, country, rating, clocks)
- disable/enable auto update
- go to last move automatically after update (when new move is received, and user is watching some other position)
- not to go to last move automatically
- play game automatically with given delay from move 1
- play game automatically with given delay from given move (currently active move)
- play given variant
- set animation speed
- set colors for boards, highlight squares, moves panel, analysis panel
- set clock format hh:mm:ss or mm:ss
- set clocks switch from hh:mm:ss to mm:ss or back depending on clocks values
- custom pieces sets switching
- custom metalanguage text for all interface items.

Server module description:
Functionality
- read PGN file(-s) remote at given interval
- analyze each move with proposed chess engine and provide output to client in a format of -+ (score) format and list proposed ‘lines’ (moves)
- store data for later use for each move
- provide data about the clocks and time spent
- create analysis database for each ‘seen’ position by several chess engines used
- allow split of the work to several servers e.g. via split of analyzed games for faster analysis
- receive and save data analysis by commentators

Requirements:
- minimal response times between server and client not more than 200ms
- analysis depth of 18 or higher
- maximally fast processing by using multiple threads and cores
- high scalability to analyze from 1 to 1000 boards simultaneously
- easy to make setup of software for fast hardware migration from server to several servers
- support large number of connections (read or push depending on technology selected by developer) (as high as 10M concurrent)
- change size by settings and mouse/mobile dragging
- show clocks for white and black with running time
- highlight moves squares
- provide animation of moves
- allow sound on moves update
- show current position on the board upon click on any move in a moves list of variations
- notifies if moves shown is an actual move or move played from variant proposed by chess engine
- shows time spent for each move
- shows name, country, rating, photo of black and white players
- calculates clock values even if clock data stops in PGN file

- constant configuration reading and assigning states to ones given in configuration
- loads games dynamically via ajax
- several instances running on one page
- use javascript engine for analysis - not take analysis from server module
- switch board (rotate board)
- show list (icons) of taken pieces
- show advantage panel
- play moves
- replay whole game
- go to the first move
- show which color is to move
- show result